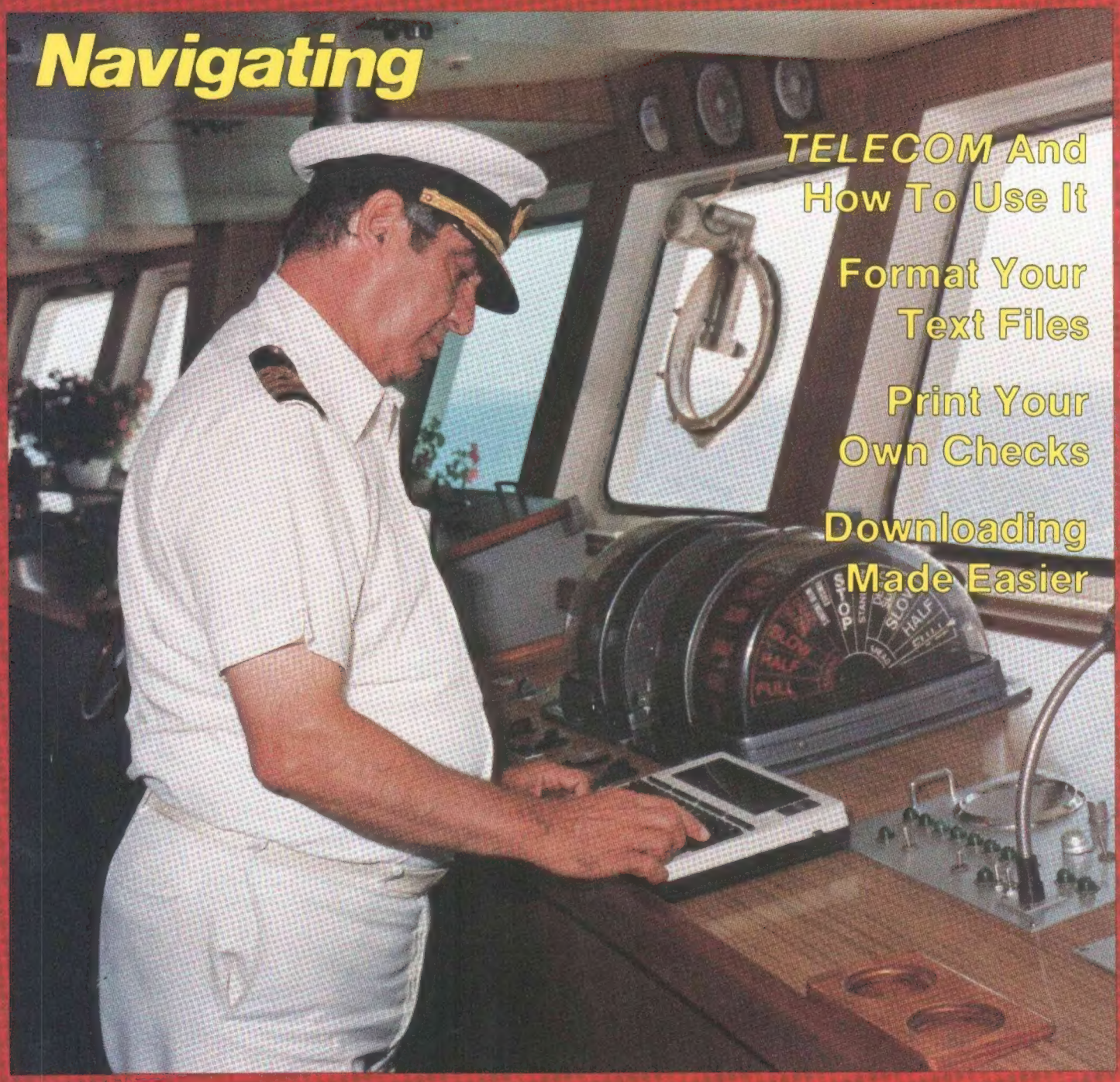
October 1983 $3

PCM™

Vol. 1 No. 4

THE PORTABLE COMPUTING MAGAZINE

FOR THE TRS-80 MODEL 100® . . . FROM THE PUBLISHERS OF *THE RAINBOW*



0 44254 00002 10

First with software for the Model 100

BUSINESSPAK+

For the Radio Shack TRS-80 Model 100

INTRODUCTORY PRICE! $89.95 Complete package includes manual and six cassettes!

six programs designed for the businessperson

- each on its own cassette
- 120 page manual in simple language

With BUSINESSPAK+ the Model 100 is virtually a complete tool for the manager or executive. Picture how this added capability will help you.

WRITE+: Instead of merely a text preparation device the Model 100 becomes a real word processor. Margins . . . side, top and bottom. Title page, topic headings, page numbers, and even headers and footers. You can even right justify and print multiple copies. It has over 36 features and countless format possibilities.

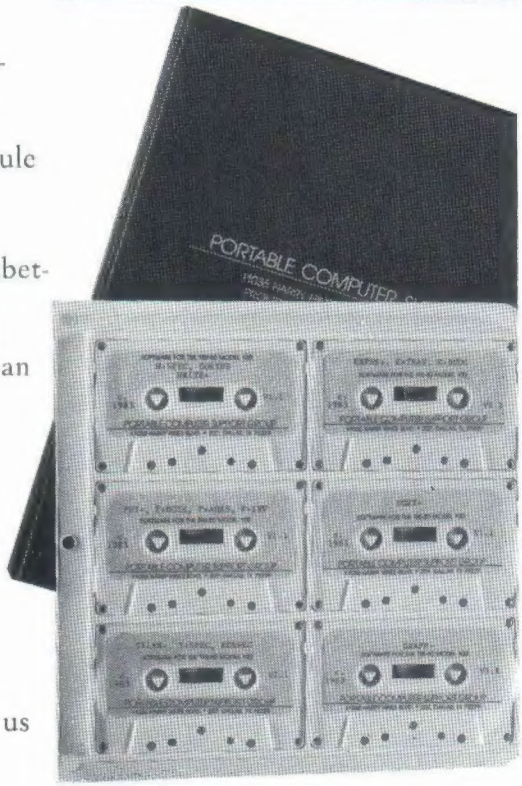
EXPNS+: By simply entering expenses into eighteen categories, you get a twelve day spreadsheet, giving totals for each day, and category. Names of rows and columns are instantly changeable, and the sheet becomes a budget or even a financial report. Replication across the columns, cumlative totals, and automatic growth or decline rate projection, make EXPNS+ an excellent planning tool. And, all the math is built right in.

GRAPH+: Prints out Bar, Line and Pie graphs of any EXPNS+ report on the DMP-100 dot matrix printer. Shows the percents on the pie chart and values on the others.

TELEX+: From any phone send any text file to any Telex machine in the world. Just enter the receiving Telex number. Send Mailgrams for next day delivery to any address at reduced rates through the Action Telex system.

PUT+: Lets you list any information in an organized fashion. It prompts for each entry and very rapidly creates addresses or schedule items and even inventory.

SORT+: Allows you to sort any list you compile with PUT+ alphabetically or numerically from any category in the entry. It can sort large files up to 20 times faster than an all Basic program.



MASTERCARD/VISA/COD

PCM CERTIFICATION SEAL

PCSG provides hotline software support for the Model 100. Give us a call at 1-214-351-0564.

PORTABLE COMPUTER SUPPORT GROUP

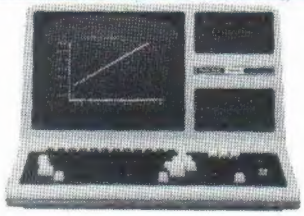
11035 Harry Hines Blvd. No. 207, Dallas, Tx. 75229

From Computer Plus to YOU . . .

PLUS *after* PLUS *after* PLUS

Model 100 8K $679
Model 100 24K $835

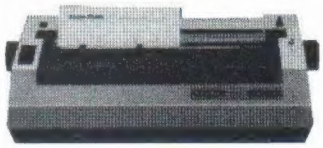
Color Computer II 16K $185
w/16K Ext. Basic $245
w/64K Ext. Basic $305



Model IV 16K $849
Model IV 64K
2 Disk & RS232 $1699

CITOH Prowriter $375
CITOH Prowriter II $649

CoCo Drive 0 $329
CoCo Drive 1 $235



DWP210 $629

BUY DIRECT *Here are just a few of our fine offers . . . call TOLL FREE for full information.*

COMPUTERS	
Model 12 64K 1 Drive	$2699
Model 12 64K 2 Drive	3375
Model IV 16K	849
Model IV 64K	
2 Disk & RS232	1699
Color Computer II 16K	185
w/16K ext. basic	245
w/64K ext. basic	305
Pocket Computer 2	165
Model 16B 1Dr 256K	4249
Model 16B 2Dr 256K	4915
Model 100 8K	679
Model 100 24K	835
MODEMS	
Lynx MI/MIII	235
Hayes Smartmodem II	235
Hayes Smartmodem 1200	565
Novation Smartcat 1200	459
Novation J-Cat	125
R.S. AC-3	129
R.S. Modem I	89
R.S. Modem II	160
R.S. DC-1200	565
PRINTERS	
Smith Corona TPI D.W.	469
Silver Reed EXP500 D.W.	430
Silver Reed EXP550 D.W.	665
Daisy Wheel II	1745
DWP210	629
DWP410	1159
CGP115	159
DMP100	315
DMP120	395
DMP200	520
DMP400	1010
DMP500	1219
DMP2100	1745
Gemini 10X	315
Gemini 15	425
CITOH Prowriter	375
CITOH Prowriter II	649
Okidata	CALL
Epson	CALL
DISK DRIVES	
R.S. Model IV Drive 0	515
Color Computer Drive 0	329
Color Computer Drive 1	235
Primary Hard Disk M12	2689
Primary Hard Disk MIII	1799
ETC.	
CCR-81 recorder	52
C.C. Joysticks (pair)	22
16K Ram Chips	25
64K Ram Chips	75
8K Par/Par Microfaser	135
Parallel printer cables are available for most computers.	
SOFTWARE	
CoCo FHL Flex D.O.S.	69.95
R.S. Software 10% off list. Send for listing of brand name software.	

We have the lowest possible Fully Warranteed Prices AND a full complement of Radio Shack Software.

Prices subject to change without notice.
Not responsible for typographical errors.
TRS-80 is a registered trademark of Tandy Corp.

AMERICAN EXPRESS · VISA · MasterCard

TOLL FREE
1-800-343-8124

computer plus

P.O. Box 1094
480 King Street
Littleton, MA 01460
617-486-3193

Write for your free catalog

SINCE 1973 —

Contents

PCM

The Portable Computing Magazine

Editor Lawrence C. Falk

Managing Editor Courtney Noe
Senior Editor Jim Reed
Assistant Editor Susan A. Remini
Submissions Editor Jutta Kapfhammer
Editorial Assistants Valarie Edwards, Suzanne Kurowsky, Noreen Morrison, Joyce Palgy
Contributing Editors Dan Downard, Robert Frowenfeld, Richard White
Art Director Jerry McKiernan
Assistant Art Director Sally Nichols
Designers Peggy Henry, Neal C. Lauron
Advertising Manager Charlotte Ford
Advertising Assistant Lynda Wilson

General Manager Patricia H. Hirsch
Assistant General Manager for Finance Donna Shuck
Customer Service Manager Ivanka Kleier
Research Assistants Deidra Henry, Tanya Holder, Monica Wheat, Wanda Perry
Dispatch Mark Herndon
Transportation Wendy Falk

PCM is represented in the Eastern United States by Garland Associates, Inc., P.O. Box 314, S.H.S., Duxbury, MA 02331, (617) 934-6464 or 934-6546. Advertisers east of the Mississippi may contact them for further information.

PCM—The Portable Computing Magazine is published every month of the year by FALSOFT, INC., 9529 U.S. Highway 42, P.O. Box 209, Prospect, KY, 40059. Phone (502) 228-4492. **PCM — The Portable Computing Magazine** and the PCM logotypes are ® Trademarks of FALSOFT, Inc.

Subscriptions to **PCM — The Portable Computing Magazine** are $28 per year in the United States. Canadian and Mexican rates are U.S. $35. Surface mail to other countries is U.S. $64, air mail U.S. $85. All subscriptions begin with the next available issue.

Payment accepted by VISA, MasterCard, American Express, Cash, Check or Money Order in United States currency only.

Limited back issues are available. Please see notice for issues which are in print and costs. Payment accepted by VISA, MasterCard, American Express, Cash, Check or Money Order in U.S. currency only.

Features

Departments

Advertisers Index

OUR COVER this month features the Portable Computer and Captain Michael Benas of the *T.S.S. Stella Solaris* as his ship passes through the Dardanelles en route to Athens from Istanbul.

'Fiddling' With Publications A Tradition

As you no doubt know, *PCM — The Portable Computing Magazine* isn't the only magazine we publish. In fact, now there are three.

The third one is called ScoreCARD, a fan magazine for University of Louisville athletics. And, while *PCM* and its sister computer publication, *the Rainbow* caused little interest in the Louisville, Kentucky, area where we hang out, ScoreCARD caused something of a stir, since it is, essentially, a local magazine devoted to local people.

Well, we got some amount of local publicity, of course, and the writers (especially the sports writers) were surprised to discover that we published two other magazines.

"How do you go about *starting* a magazine?" was the most frequently asked question we got.

In some ways, this reminded me of the response Tevya, the hero of *Fiddler On The Roof* answered his self-asked question when he wondered why the fiddler stayed on the roof if it was so dangerous. "I don't know," said Tevya. "But it's a tradition."

In the last six months, we've started two magazines, so maybe that's a tradition, too. And all of this leads me to be able to proudly point out to you that, yes, *PCM* gets a little bigger this month and is expected to be getting even bigger in the future.

There are two reasons this is happening. The first, obviously, is there are more advertisements this month — which means it is easier to *pay* for those extra pages. But the second reason is the more important one — both to us and to you.

We're starting to get more and more submissions to our editorial columns, more people wanting to do product reviews, more letters and notes with offers of help and suggestions. And, yes, more subscriptions, too.

All of that means that we're able to bring you more information in *PCM* and on your Portable Computer than we have been able to do before. To understand that fully, you need to know something about our basic philosophy of magazine publishing.

I do not believe that the majority of you are really interested in the technical aspects of your Portable Computer. Most of you bought it to do a job — be that job word processing, keeping files, making an analysis of data or some leasure activities. Usually, that is a combination of both. I think you want us to show you how to do those things — not how to open up your computer (which, by the way, voids your warranty) and solder this chip to that. I doubt many of you want to disassemble the ROM routines and the like.

Which brings me to the point (besides the welcome news that we're growing) of all this: We solicit user articles, programs and reviews. If you have nothing to review, let us know — we'll add you to the list of people who want to review for us. And, of course we'll pay for your work.

I believe that virtually every program written for a computer has been developed to serve a specific need. Most of us are beginners. In fact, the very popular Bob Albrecht, one of the most widely-read and successful writers in the microcomputer field today, characterizes himself as the "ultimate beginner." The reason, as Bob explains it, is because he's always learning something new and when he writes about something, he wants it to be clear and understandable to others who are trying to grasp something that is new to *them*.

The point of all this is that it seems to me that, sometimes, potential contributors are "dazzled" by the programs they see in magazines and feel that *they* could never come up with anything so good, so they just never send in a program they've written. Often, I've found that some of the best programs are those which have to be "coaxed" out of a writer.

One of our contributing editors in *the Rainbow* is an excellent case in point. He wrote me a letter very early on in the magazine's history, saying he had a program but didn't know if it was worth sending. He asked me whether I'd be interested.

When I wrote back saying I would, he wrote back again, saying that he'd thought about it and that, on reading *the Rainbow*, he decided it was nowhere near up to the standards of the programs we were printing. He thanked me for my time.

I called him and convinced him that he ought to send it in anyway. It was a good program. We ran it and, later, ran a couple more. Now, he contributes to *the Rainbow* on a regular basis and his programs are some of the best we print.

I'm trying to emphasize that we would welcome your contributions very much. In writing any program you've done, you have obviously addressed a need you have — whether that be for business use, for entertainment or just to learn something. You can share that with thousands of other people who also have a Portable Computer — and get paid for it, too.

I hope that you will consider sending us your programs and information you've learned about the Portable Computer. You'd help your fellow PoCo owners, help yourself and help us as well. And, who knows, a year from now you might be one of the "names" in the microcomputer world!

From the start, *PCM* has been an "International" publication, because we have a great many subscribers and retail outlets in Canada. Now, though, you can also get a special Australian version of the magazine "Down Under." Our agent in Australia reports that sales of

(continued on page 38)

A BASIC START

A new column on BASIC for the M100 by RICHARD WHITE, *PCM* Contributing Editor.

I see two types of Model 100 owners. There are quite a few who are computer conversant and who realized the strengths of the machine immediately. Many of these folks had their money down within the first few weeks. Some had been thriving on old Model I's and had not seen the value of upgrading. That is, until Model 100. Others had a number of machines, but could not resist the next major advance. Many of these owners need a column on BASIC like they need three left shoes. This column is addressed to those having little programming experience, but who want to start using the BASIC in their machines. We will talk about things that are in the manuals and occasionally dig into things that are not. And we will digress into some non-code issues like programming practices and program structure.

Of course, you could start to learn BASIC by using the manual that comes with your machine. At least, you will need to know where everything is located in the manual. But, the Model 100 Manual is not a good text to start from scratch with. Its BASIC section is really not much more than an annotated listing of the available commands and other data. It looks like a good reference, but hardly a beginner's text. Far better would be the Color Computer manuals which take a step at a time approach. These can be purchased at your local Radio Shack store. Alternately, you might choose from other BASIC texts at Radio Shack, B. Dalton's or other book sellers. Don't worry about whether your Model 100 language is somewhat different. There are refinements and additions and the screen locations differ, but the basic commands, functions and operators you need to learn at the beginning are the same. And you always have your M100 manual to check to see if things are the same.

There are some things you should know early. One is how to save programs to tape and load them back into your machine. Even if you have 32K, you will quickly fill memory without magnetic storage. If you have access to another computer, you could dump to that machine and save to disk. I suspect tape will prove less hassle in many instances. Of course, when traveling, a recorder can be carried in the suitcase and files dumped in the evening making room for the next day's business and notes. If you don't have access to another computer, you have no choice but to use a tape recorder.

A word about tape reliability. The Color Computer was the first computer to have a really viable tape interface. I used tape for two years before getting my disk and did not feel deprived. The M100 has a very similar interface so it will perform equally well. Perhaps the keys to satisfactory cassette operations are to use short (C-10 or C-20) tapes, always save twice, and use a recorder with a tape counter so you can write on the cassette itself the start and end location for saves

of each program. I have an adage: *fools save only once while cowards save three times.* If you cannot find C-10 cassettes, check the ads in this issue and order by mail. I cannot argue too strongly about writing the contents and counter location on the cassette label. If you do this regularly, you will find you can put two to four programs per side and access any one without particular stress. As you do programming, it is handy to have a SAVEing utility built into your program. Yes, I know about function key 3 and find the following method and code much better. I generally use Line 10000 (easy to remember).

```
10000 INPUT"WANT TO RUN PAST
LEADER";I$ : IF I$="Y" THEN
MOTORON : FOR X=1TO 5000 :
NEXT

10010 FOR S=1 TO 2 : CSAVE
"YOURPROG" : MOTORON : FOR
X=1 TO 500 : NEXT : NEXT :
MOTOROFF
```

Other than being handy, we can dissect this program bit and see a lot about the basics of BASIC. First you use this by typing RUN 10000 or GOTO 10000. Ah so, we do not always have to start at the beginning of a program. Indeed, you could have a number of programs in a file, and RUN the one you want by its starting line number. Of course, in most cases it will be handier to have programs as separate files in memory and run them from the main menu. Not in this case, since you want to save the current file in a specific way to tape. You also cannot use the F4 key since that always RUNs a program starting at the lowest line number.

There is a lot that can be learned about programming concepts and BASIC in those few lines of code. So, I will embark on a long winded discussion and hope you pick up some ideas along the way. Most tapes have leaders, and many times you will need to position the tape past the start of the leader. Why not have the computer remind you and then run the tape past the leader if that's what is needed? We start with INPUT"WANT TO RUN PAST LEADER Y/N";I$. The computer will print the text between the quotes and then wait for you to enter a "Y" or "N" and press ENTER. INPUT is neat in that the programmer can put on the screen a message about what response is needed and get that response in one command. Here, INPUT is looking for a character or string entry, but it can be used to get numeric data as well if a numeric variable is used. For example INPUT"ENTER A NUMBER";X.

> "Reading a spaghetti program is rather like reading an upside down roadmap . . ."

So something was keyed and the enter key pressed. Now the computer must make a decision. There are a number of ways to make a decision in BASIC. IF . . . THEN . . . ELSE is probably the most used. Since we can string a number of commands in one line of BASIC, separating them by colons, we will put our test in line 10000. IF I$="Y" THEN MOTORON : FOR X =1 TO 5000 : NEXT. IF makes a true false test. If I$="Y" is true the code after THEN is executed. If I$ equals anything else, the test proves false and control goes to the next BASIC line. If you do want to run past the leader, MOTORON turns on the tape recorder. You did have the play and record buttons down? To run past the leader, the cassette must run 12 to 15 seconds. We need a timer. But, it takes time to run a program. If we tell the program to run around in circles (loop if you want to be technical) so many times, it will use a predictable amount of time. That is what FOR . . . TO . . . NEXT does in this case. The M100 starts at X=1 and goes to NEXT. NEXT checks if X=5000 or is greater than 5000. If not, it adds one to X and sends it back to the FOR statement. This is the simplest possible FOR . . . TO . . . NEXT, just to give you beginners the idea.

Both IF . . . THEN and FOR . . . TO . . . . NEXT control what the computer does so they are called control structures. There are thousands of ways they can be used and a beginning programmer could do worse than spend the time to learn their ins and outs. And that done, you will learn even more as you use them.

In line 10010 we are at the meat of the matter. This saves the program to cassette twice and puts some blank space between saves. FOR S=1 TO 2 : CSAVE"YOURPROG" makes the first save and there is a FOR . . . TO again. BASIC always goes through a FOR . . . TO at least once so you would have gotten one save even if there were -100 in place of 2. Moving on, MOTORON : FOR X=1 TO 600 : NEXT : NEXT : MOTOROFF. I said we wanted to leave a space between CSAVEs so MOTORON turns the cassette motor on and FOR X =1 TO 600 : NEXT does a time delay. There are two nexts in a row, but M100 knows which FOR . . . TO it's dealing

with and won't get mixed up. At this point S = 1 so the second NEXT sets S = 2 and sends the program back to the first FOR . . . TO to CSAVE the program again. A space is left at the end of the second save and MOTOROFF turns the recorder off.

In this bit of program or code as we like to call it, to increase the fog index, we prompted the user for an input, got it, tested it and caused the computer to do four or five separate functions depending on that test. We met two BASIC control structures in simple form and used one three times. Not bad for two lines of code. It even makes sense when divided into little pieces. Perhaps the secret to programming is to "divide and conquer."

Perhaps? Nay, it's a certainty. If you think of your programs as functional little pieces, you will do a lot better. Pascal, another programming language, was written for use in teaching programmers to structure their programs. In fact, you cannot write in Pascal without adhering to strict structural rules. In BASIC you have a choice of writing free spirited spaghetti-like programs or relatively structured ones. Reading a spaghetti program is rather like reading an upside down roadmap under a new moon at midnight. New programmers seldom know where they are going with a program and halfway through may get bogged down figuring where they have been as well. Such is part of the learning process, so don't lose heart.

Program structure means program organization. Some languages require that variables, files, data structures and other attributes be declared at the beginning. Generous use of remarks is encouraged and specific indentation formats are strongly suggested. Structuring also deals with what should be in subroutines, what should be in the main procedure and how the procedures flow. Much of this is optional in BASIC, some is just good practice in any language.

Clarity should be as important a goal in BASIC as it is in other languages. There are a number of ways to write clear programs. I use the procedures that follow and find them valuable.

1) Define specific program functions and put the code for each function in its own module with introductory REMs. Assign a specific set of lines to a module. Blocks of 100 lines are convenient and will meet most needs. You will always know a module begins at an even hundred and can go right to the one you want.

2) Minimize looping back. The procedure should flow from start to end and loop only to repeat the routine or a portion of it.

> "Perhaps the secret to programming is to 'divide and conquer.' "

3) Use IF . . . THEN . . . ELSE to minimize jumping forward. Many times all the code for the choices can be contained after THEN and ELSE on one line so that the procedure can continue on the next line. Some BASICs don't have ELSE. Apple doesn't and Apple owners pay extra for the lack. To illustrate, which is clearer to you?

```
10 IF X=0 THEN PRINT "FALSE":
X=1 ELSE PRINT"TRUE"
20 END
or
10 IF X=0 THEN 30
20 PRINT "TRUE": X=1: GOTO40
30 PRINT"FALSE"
40 END
```

4) Put all subroutines in one of two places. Frequently called subroutines which include those that affect program speed should be at the front of the program. I reserve lines 5 to 99 for these. These should not be more than three or four lines. Line number spacing of two is good. Putting these here serves two functions. First the computer finds them quickly when speed counts. You also save bytes since the number in the GOSUB is only one or two bytes long. Infrequently called routines, particularly program initialization code should be at the end of the program. Each time a subroutine is called or the computer is sent to a specific line, it starts at the beginning of the program and searches until it finds that line. The fewer lines it passes to find the needed line the faster the search will be. It follows that speed is compromised if the computer is continually searching over code it has used and will not use again. Clarity comes from having only two places to look for subroutines. The one exception that I make is to place a major subroutine at the end of the module that calls it when only that module uses it.

5) The same reasoning used above also applies to the ordering of main program modules provided they are called separately. Those used most frequently are put in front of the occasionally used ones. In a file program, the input module is used far more than the save to tape module and should come to the front. Modules that are used in order should be placed in order in the program.

6) Menus should be placed where they are used in the program. A menu's text provides valuable information on the branching of the program that follows. If menu choices are numbered from one up in sequence, the ON I GOTO XXX, YYY,ZZZ or ON I GOSUB XXX, YYY,ZZZ commands can be used. It is easy to read a listing, see which number corresponds to the code block you want and then drop down to the ON I statement and count across to find the target line number. Memory conservation sometimes forces use of strings defined elsewhere in the program in menu text. When this is done the advantage of reading the menu code to know what options are there and where to go for their code is lost. Programming is a series of compromises. It is up to you to know what your options are and to make sure that your choices are educated ones and not guesses or blind repetition of past practices.

> "I have an adage: *fools save only once while cowards save three times.*"

Though I have discussed program structuring for purposes of easy trouble shooting and modification, the structure proposed had program speed firmly in mind as well. Memory use was another consideration. Clarity, memory and speed are like three corners of a triangle. You cannot be at all three points at once. You can make choices that shorten the sides of the triangle and get close to where you want to be. It takes careful thought and planning at the start so you don't end up redoing too much. Don't be upset when you don't get what you want the first or second time. Remember, all good commercial programs have version numbers and Version 1.0 is the first one offered for sale, not the first one of the development process. PCM

Here is a super market program to take with you when you go . . .

Traveling To The Aisles

By Robert Frowenfeld
PCM Contributing Editor

We're going to leave the world of business ratios and economic analyses to concentrate on something a little bit closer to home. As a matter of fact, as close to home as your grocery store. This month, we're going *On The Road* with the Model 100 to assist us in doing our food shopping. Just unplug your PoCo, stroll on down to the local supermarket, place your TRS totable in the top of your shopping cart, and get ready to simplify your shopping. A word of caution here before we get started. There may not be enough room in your shopping cart for both your Model 100 and junior, so if you really want to make good use of your computer, you may have to leave the apple of your eye alone to walk down the aisles and disrupt an otherwise orderly store.

Using the first four letters of the word *grocery*, I've decided to name this program *GROC.BA;* of course you can save it under any name you want on your Model 100 as long as you use the *.BA* extension. *GROC* was designed to perform two basic functions. The first is to keep track of all your purchases and to record them in such a manner so that you can easily review your expenditures when you get home. By classifying your purchases into specific categories as well as letting you type in a description for each item, the program can make a simple job of reconciling your cash register tape. Furthermore, as you'll see later, it can also let you know how much of your food dollar is being spent on various categories of items. The second function of *GROC* is to help you comparison shop for the best prices. There's a built-in unit cost price calculator which will help you decide which items are the better buys.

So, let's get started. The main menu (see Figure #1) lists all the functions available. The first option is the *Compute Unit Price* function. By pressing 1, you will be prompted by the Model 100 to enter a price formula. Here, you have two choices. You can type in an equation to give you the unit cost of an item, or you can enter a formula to have the Model 100 give you your total price for an item sold in quantity. As an example of the unit price function, let's say you are confronted with two sizes and prices of peanut butter, and you don't know which is the better value. The smaller size is $1.49 for 18 ounces, and the larger is $2.19 for 25 ounces. For the first size, type in the equation: *18/1.49* and press ENTER; the program will respond with your unit cost of $0.083; this represents the cost per ounce of the smaller size. For the larger size, enter *25/2.19* and *GROC* will respond with $0.088, indicating that the smaller size is really the better buy.

The second function of the unit price calculator is to determine the cost of items that aren't priced by the piece. As an example, oranges are selling for $1.79 per dozen, but you only want to buy five of them. Simply enter *5@ 12/1.79,* and your trusty shopping companion will respond with your cost of 0.75 to indicate your cost of 75 cents. This will be helpful when you need to enter the price paid for your purchases in the following section.

Enter Purchase (Figure 2) allows you to record your purchases on your Model 100 so that you can later review them to help reconcile your cash register receipt. When entering an item that is being purchased, you must assign a category number when prompted with the message *Select Item.* Seven different categories have been allocated to *GROC.* They are: Bakery, Dairy, Deli, Groceries, Meat, Produce, and Taxable Item. The titles for these headings can be found on line 60 in the program. If you desire, they can be changed, but don't change the last one — *Taxable Item* since the program adds in sales tax at the end based on the total of items selected for this group.

Once you've selected a category, you are prompted to enter the price. If you have to go back and compute the price using the unit price compute function, just press ENTER; the program will take you back to the *Select Item* prompt. From here you press 8 to return to the main menu and compute your actual cost. Once you've entered the price you'll be asked to furnish *GROC* with a description of your purchase. Since we're storing text here, I recommend that you keep your descriptions brief so as not to use up too much memory and/or string space.

Now that you have your purchases entered, you can do one of several things. First, I suggest that you save a record of what you've bought by selecting option #4, *Save Purchase File.* This will record all your purchases in a RAM file named *PURCH.DO.* In fact, it is recommended that you periodically save

your data to this file, just in case you have to shut off your Model 100 for a while to explain to the store manager why your little boy (angel that he is) won't stop playing with the lobsters in the tank. Once you get home and you want to review and/or list your purchases, you can load this file into *GROC* by using option #5, *Load Purchase File.* Note: the only time you have to load the purchase file is when you have had the machine turned off or, for some reason, you stop or exit the program and want to resume using it where you left off. If you just want to save the data and continue, you don't have to reload it.

This brings us to the last function of *GROC,* option #3: *Purchase Summary.* This allows you to list and/or print a summary of all your purchases. The listing itemizes each purchase by indicating the price paid and description of each item. At the end of the listing is a subtotal, a total for the taxable items, the amount of sales tax paid, and a grand total. The program has been configured for a sales tax rate on taxable items of 5 percent. If your sales tax rate is different, you will have to change line 30. For example, if you live in New York City, line 30 will read *30 TR = .08125* since your tax rate is 8¼ percent.

At the end of the list is a recap of the dollar amount and percentage of your grocery bill paid for each of the seven purchase categories.

One additional note, this program was originally set up to handle 100 items. If you feel you'll need more capacity — no problem. Just change the arrays dimensioned in line 40 to accommodate a larger number of items; you might also want to change the cleared string space in line 10. Well, until next time — *Bon Appetit!*

Figure 1

```
PCM Shopping Companion
     Select:
     1  Compute Unit Price
     2  Enter Purchase
     3  Purchase Summary
     4  Save Purchase File
     5  Load Purchase File
     6  End Program
```

Figure 2

```
              Enter Purchase
  1  Bakery              5  Meat
  2  Dairy               6  Produce
  3  Deli                7  Taxable Item
  4  Groceries           8  Exit

    Select Item:
```

Figure 3

```
 Item #      Price      Description
    1         1.25      DONUTS
    2         1.49      MILK - 1/2 GAL
    3         0.79      ALUMINUM FOIL
    4         1.50      ORANGES (8)
    5         0.99      CREAM CHEESE
    6         0.69      CANNED PEACHES
    7         4.59      T-BONE STEAK
    8         0.79      BREAD

 Subtotal:              $ 12.09
 Taxable Total:         $  0.79
 Sales Tax @ 0.050%     $  0.04
 Grand Total:           $ 12.13

 CATEGORY                 TOTAL       PERCENT
 Bakery                 $  2.04        16.82
 Dairy                  $  2.48        20.45
 Deli                   $  0.00         0.00
 Groceries              $  0.69         5.69
 Meat                   $  4.59        37.84
 Produce                $  1.50        12.37
 Taxable Item           $  0.83         6.84
```

The listing:

```
1 GOTO 10
6 LINEINPUT IN$:IF IN$<>""THEN Y=ASC(IN$
)
7 X=VAL(IN$):RETURN
10 CLEAR 2000:DEFINT I-N:DEFSNG C-H,O-Q,
S-T,V-Y:DEFSTR A-B,R,U,Z
20 R=CHR$(27)+"p":U=CHR$(27)+"q"
25 F1$="$###.##"
30 TR=.05
40 IX=0:DIM TY(100),PR!(100),DE$(100)
50 DATA "Compute Unit Price","Enter Purc
hase","Purchase Summary","Save Purchase
File","Load Purchase File","End Program"
:FOR I=1 TO 6:READ M0$(I):NEXT I
60 DATA"Bakery","Dairy","Deli","Grocerie
s","Meat","Produce","Taxable Item","Exit
":FOR I=1 TO 8:READ M1$(I):NEXT I
99 GOTO 1000
100 'unit price
```

```
110 GOSUB 600
120 PRINT@80,"Enter price formula: ";:GO
SUB 6:IF IN$=""THEN 1000
130 J=INSTR(IN$,"/"):IF X=0 OR J=0 OR J=
LEN(IN$) THEN GOSUB 650:GOTO 100
140 K=INSTR(IN$,"@"):IF K=LEN(IN$) THEN
GOSUB 650:GOTO 100
150 IF K<>0 THEN QU=VAL(MID$(IN$,K+1))
160 Y=VAL(MID$(IN$,J+1)):IF Y<=0 THEN GO
SUB 650:GOTO 100
170 IF K=0 THEN C=Y/X:PRINT@160,"Unit Co
st = ";USING"$##.###";C:GOTO 190
180 IF K<>0 THEN C=X*Y/QU:IF C*100<>INT(
C*100) THEN C=INT(C*100+1)/100
182 C=C
185 PRINT@160,"Total Price ="USING"##.##
";C
190 GOSUB 610:GOTO 100
200 'enter purchase
210 GOSUB 1100
220 PRINT@280,STRING$(38," ");:PRINT@245
,"Select Item: ";:IN$=INPUT$(1):X=VAL(IN
$):IF X<1 OR X>8 THEN 220 ELSE J=X:PRINT
J;
230 IF J=8 THEN 1000
240 P=6+J*40:IF J>4 THEN P=P-140
245 PRINT@P,R;M1$(J)U;
250 PRINT@265,"Price: ";:GOSUB 6:IFIN$="
" THEN PRINT@P,M1$(J);:GOTO 220
260 PR=X:IX=IX+1:PR!(IX)=PR!:TY(IX)=J
270 PRINT@280,STRING$(38," ");:PRINT@280
,"Description: ";:GOSUB 6:IFIN$="" THEN
270 ELSE DE$(IX)=IN$
280 GOTO 200
300  purchase summary
305 GOSUB 600
310 P=0:PRINT@120,STRING$(39," ");:PRINT
@127,"Send output to Printer (Y/N): ";:G
OSUB 6:IF IN$="" THEN 1000 ELSE IF IN$="
Y" OR IN$="y" THEN P=-1 ELSE IF IN$<>"N"
AND IN$<>"n" THEN 310
315 CLS:TT=0:FOR I=1 TO 7:T(I)=0:NEXT I
320 PRINT"Item #     Price     Descripti
on":IF (P) THEN LPRINT"Item #     Price
    Description
325 FOR I=1 TO 100:IF PR(I)=0 THEN 345
330 PRINTUSING" ###       ##.##     ";I;
PR!(I);:PRINT DE$(I)
335 IF (P) THEN LPRINTUSING" ###       #
#.##     ";I;PR!(I);:LPRINTDE$(I)
340 T(TY(I))=T(TY(I))+PR!(I):TT=TT+PR!(I
)
345 NEXT I
350 PRINT:IF (P) THEN LPRINT
355 PRINT"Subtotal:           ";USINGF1$;
TT:IF (P) THEN LPRINT"Subtotal:
 ";USINGF1$;TT
360 PRINT"Taxable Total:     ";USING"$##
#.##";T(7):IF (P) THEN LPRINT"Taxable To
tal:     ";USING"$###.##";T(7)
365 TX=TR*T(7):PRINT"Sales Tax @ "USING"
#.###";TR;:PRINT"% $";USING" ##.##";TX
366 IF (P) THEN LPRINT"Sales Tax @ "USIN
G"#.###";TR;:LPRINT"% $";USING" ##.##";T
X
370 TT=TT+TX:T(7)=T(7)+TX:PRINT"Grand To
tal:       "USINGF1$;TT:IF (P) THEN LPRI
NT"Grand Total:       "USINGF1$;TT
372 PRINT:IF (P) THEN LPRINT
375 PRINT" Press any key for summary by
category";:A=INPUT$(1):PRINT
378 PRINT"CATEGORY"TAB(20)"TOTAL      PE
RCENT"
379 IF (P) THEN LPRINT"CATEGORY"TAB(20)"
TOTAL      PERCENT"
380 FOR I=1 TO 7
385 PRINT M1$(I)TAB(19)USINGF1$;T(I);:PR
INT"     "USING"###.##";T(I)/TT*100:IF (
P) THEN LPRINTM1$(I)TAB(19)USINGF1$;T(I)
;:LPRINT"     "USING"###.##";T(I)/TT*100
390 NEXT I
395 PRINT:GOSUB 610:GOTO 1000
400 'save
410 GOSUB 620:OPEN ID$ FOR OUTPUT AS 1
415 PRINT@173,"Saving Data ...";
420 FOR I=1 TO 100:PRINT#1,TY(I);PR!(I);
CHR$(34)DE$(I);CHR$(34)" ";:NEXT I:CLOSE
:GOTO 1000
450 'restore
460 GOSUB 620:OPEN ID$ FOR INPUT AS 1
465 PRINT@173,"Loading Data ...";:IX=0
470 FOR I=1 TO 100:INPUT#1,TY(I),PR!(I),
DE$(I)
475 IF TY(I)<>0 THEN IX=IX+1
480 NEXT I:CLOSE:GOTO 1000
500 'end
510 CLS:MENU
600 CLS:PRINT@0,STRING$(20-LEN(M0$(FX))/
2," ");M0$(FX);:RETURN
610 PRINT@285,"Press any key to continue
";:A=INPUT$(1):PRINT@285,STRING$(30," ")
;:RETURN
620 GOSUB 600:ID$="PURCH.DO"
625 RETURN
650 PRINT@255,R" ERROR "U;:FOR I=1 TO 10
00:NEXT I:RETURN
1000 'main menu
1010 CLS:PRINT@10,"PCM Shopping Companio
n"
1020 FOR I=1 TO 6:PRINT@40*(I+1),TAB(5),
R;I;U" "M0$(I);:NEXT I
1030 PRINT@55,"Select: ";:IN$=INPUT$(1):
X=VAL(IN$):IF X<1 OR X>6 THEN 1030 ELSE
FX=X:ON X GOTO 100,200,300,400,450,500
1100 'purchase menu
1110 CLS:GOSUB 600:FOR I=1 TO 4:PRINT@I*
40+2,R;I;U" "M1$(I):NEXT I
1120 FOR I=5 TO 8:PRINT@40*(I-4)+22,R;I;
U" "M1$(I);:NEXT I
1130 RETURN
```

PCM

Have you ever wondered how nice it would be to have your computer print checks for you? Of course you have! It's so easy (of course). You set your computer in front of you and you enter the date, the amount, and the payee and your trusty micro prints out a check. Simple, right? I mean, what could be easier? Well . . . it's not exactly that simple. It's not so simple because of the line where you are supposed to write out the amount of the check in words. You know — $347.12 becomes THREE Hundred FORTY-SEVEN Dollars and $^{12}/_{100}$, etc. Now how would you get your micro to do this? As you may have guessed, the answer is in the accompanying listing. With a few DATA statements and a little clever programming, I've come up with a nifty routine to solve this problem. The subroutine listed (which will be explained in detail) is capable of generating an alphabetic string for any amount from $0.01 through $99,999.99. Following are several examples. Before we discuss the mechanics of this routine let me digress for just a moment. While this program was originally written for a Model 100, it can be used without any modification on just about any other computer. Also, if you do have visions of one day keeping your checkbook on a computer (or if you are already doing so), this subroutine can be used to very easily incorporate the added feature of check printing.

Now for the explanation. Lines 50, 60, and 70 are DATA statements. These are simply the words for the various numeric amounts. Line 50 contains the numbers ONE through TEN and also the word No when the amount of the check to be written is less than one dollar. Line 70 contains the words for multiples of ten: TEN, TWENTY, . . . , NINETY. Because of the way we count between TEN and TWENTY, line 60 has been entered and contains the words for the numeric values between these two amounts. Line 80 reads these numerics words into three arrays: A1$, A2$, and A3$. Because I have defined all variables beginning with the letter "A" to be strings (see Line 10), I don't have to include the dollar sign "$" as a variable type specification.

The routine for determining the alphabetic text begins at line 110. At this point the subroutine expects a double precision variable named X# to contain the dollar amount to be examined. The reason for the value having to be in double precision is simple. The highest value this routine was designed to handle is $99,999.99, which contains seven digits. Using a single precision 4-byte variable yields only six digit accuracy. Hence, double precision is used. Because of the Model 100's default to double precision, I'll refer to the variable throughout the explanation as X and not X#.

The variable A is a string which is used to store the alphabetic interpretation of the numeric amount. In line 110 it is set to a null string. Also in line 110, the original value of X is stored in X1. Next the routine checks to see if the value of the check is less than one dollar. If it is, the string is assigned a value of "No" contained in A1(0), and execution is transferred to line 150 where the word "Dollars" will be added prior to the addition of the fractional part (cents). Lines 120, 130, and 140 all work essen-

"You set your computer in front of you and you enter the date, the amount, and the payee and your trusty micro prints out a check. Simple right? . . . Well . . . It's not so simple because of the line where you are supposed to write out the amount of the check in words. You know — $347.12 becomes THREE Hundred FORTY SEVEN Dollars and 12/100, etc. Now how would you get your micro to do this? . . . With a few DATA statements and a little clever programming, I've come up with a nifty routine to solve this problem."

Ask your dealer for the Porta™ series from Skyline—to help you work faster and better!

PortaCalc™ — Model 100 "electronic spreadsheet" with 14 column by 26 row workspace! Full use of the built-in function keys to save, load, screen print, report print, or to look behind the data at the formulas in use.

Full arithmetic operators including exponentiation, absolute value, integer, summation, and averaging. Calculations are done to 14 digits of precision and displayed up to 9 digits. User selectable decimal place from none through seven—not just one, two, or floating! Formula replication is included to allow fast creation of worksheets.

Worksheets may be saved, loaded, or merged using the computer's memory or cassette. Often used templates can be saved in memory for instant recall.

Two powerful utility programs are included with **PortaCalc** at no extra charge! **PortaDex**™ is a data exchange program that allows reformatting **PortaCalc** files into the DIF format used by **VisiCalc**. **PortaPrint**™ is an enhancement to the Model 100's text editor, adding the ability to adjust left, right, and top margins, and page length. Page numbering, headers, centered lines, flush right justification, new page control, and more.

Comes with extensive documentation in padded 3-ring binder. Includes tutorial, detailed reference section, and executive level sample templates. Fully illustrated with screen prints, examples, and hints.

PortaStat™ — Correlation, regression analysis, and descriptive statistics package, designed to be a working tool for today's businessman. Data can be input from cassette, memory, or keyboard. Interactive with **PortaCalc** data files! Thoroughly documented, includes tutorial.

PortaFin™ —Present value, net present value, future value, annuities, interest factors, loan constants, internal rate of return and more! This user-friendly package prompts you for the needed input data, then gives you the answer to your loan or investment problem! Interactive with **PortaCalc** files, allowing you to "read" cash flow data from memory. Keyboard or cassette input also available.

PortaMax™ — The powerful 'simplex' method of linear programming. Find the optimum 'mix' for a given set of constraints. Interacts with **PortaCalc** files to avoid time-wasting repetitive data entry. Tutorial, examples, and bibliography included.

PortaFolio™ — Stock and bond valuation, betas, Macaulay's duration, yield to maturity, bond swap calculations, and much more. All in an easy to use program, interactive with **PortaCalc** data files. Supplied with comprehensive instructions and reference manual.

PortaCalc

All **Porta** series programs are supplied on cassette and require 24K RAM.
Pricing — PortaCalc is **$69.95**, all others are **$44.95** each.
If ordering direct, add $2.00 shipping and handling per order.
Dealer inquires invited.

VISA MasterCard

Skyline MARKETING CORP

4510 W. Irving Park Rd. • Chicago, IL 60641 • (312) 286-0762

tially the same way. These statements set up a variable Y to equal the integer number of thousands, hundreds, and units of dollars. Each line then examines the value of Y and, if not equal to zero, calls the subroutine at line 200 to get the alphabetic expression for the number Y. For thousands of dollars, Y can have a value in the range of 1—99; for hundreds of dollars Y can be any value from 1—9; and for units, Y takes on a value of 1—99. On return from the subroutine starting at line 200, lines 120, 130, and 140 append the variable B$ to A$, and add the appropriate wording such as "Thousand" or "Hundred." At the beginning of lines 130 and 140 the value of X is decreased to handle the remaining amount. For example, say we want to print the value $4,205.67. At the beginning of line 130, the value is reduced to 205.67, and at the beginning of line 140, it is further reduced to 5.67. At line 160 X is further reduced to the number of cents remaining (in this example .67), and multiplied by 100 to make it an integer number that can be printed as a fraction over 100.

The subroutine at line 200 is what really makes this program shine! Given any number between 1 and 99, it returns the appropriate text by looking at the alphabetic arrays A1, A2, and A3. Line 200 checks to see if the variable Y is less than or equal to 10. If so, then things are easy and we simply set B$ equal to the (alphabetic) value A1(Y) and return. At line 210, we check to see if the value of Y is in the range of 11 through 19, and if so, sets B$ equal to the appropriate value found in array A2 and returns. Now comes the tricky part. We know at line 230 that the value lies in the range of 20—99 and that we will need a tens "word" and (possibly) a units "word." First we set B$ equal to the string in A3 that comprises the tens part. We do that by taking the integer part of Y divided by 10 and using the result as a pointer to the "tens" array of A3. Next, a check is made to see if the remainder of dividing by 10 is equal to 0 by using the modular arithmetic function MOD. If the result is not 0, then we add a hyphen ("—") and add the string for the units "word" from the array A1. Upon returning from this subroutine, the variable B$ now contains the appropriate text to be concatenated onto the variable A. Easy as pie!

By the time execution gets to line 150, all that is left to do is to add the appropriate fraction of 100 and we're done. When the routine returns from line 170, the variable A contains the text to be printed.

Lines 300—310 have been included to illustrate how this routine could be called by a program requiring the alphabetic text for a numeric amount. Line 300 inputs a dollar amount and checks to make sure it is within the range of $0.01 — $99,999.99; line 310 calls the subroutine and (upon returning) prints the result.

Examples:

```
Input amount: 1045.23
ONE Thousand FORTY-FIVE Dollars and 23/1
00
```

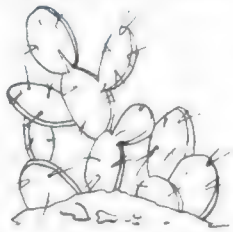
```
Input amount: 412.85
 FOUR Hundred TWELVE Dollars and 85/100
```

```
Input amount: 8046.75
EIGHT Thousand FORTY-SIX Dollars and 75/
100
```

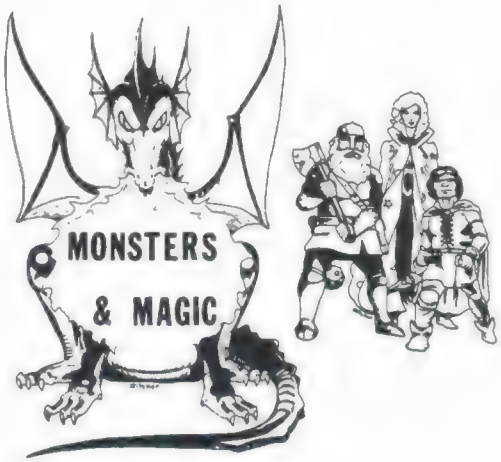
The listing:

```
10 CLEAR 500:DEFSTR A:DEFINT I:CLS
50 DATA No,ONE,TWO,THREE,FOUR,FIVE,SIX,S
EVEN,EIGHT,NINE,TEN
60 DATA TEN,ELEVEN,TWELVE,THIRTEEN,FOURT
EEN,FIFTEEN,SIXTEEN,SEVENTEEN,EIGHTEEN,N
INETEEN
70 DATA TEN,TWENTY,THIRTY,FORTY,FIFTY,SI
XTY,SEVENTY,EIGHTY,NINETY
80 FOR I=0 TO 10:READ A1(I):NEXT I:FOR I
=0 TO 9:READ A2(I):NEXT I:FOR I=1 TO 9:R
EAD A3(I):NEXT I
90 GOTO 300
100 'start
110 A="":X1=X:IF X<1 THEN A=A1(0):GOTO 1
50
120 Y=0:IF X<1000 THEN 130 ELSE Y=INT(X/
1000):GOSUB 200:A=A+B$+" Thousand"
130 X=X-Y*1000:Y=0:IF X<100 THEN 140 ELS
EY=INT(X/100):A=A+" ":GOSUB 200:A=A+B$+"
 Hundred"
140 X=X-Y*100:Y=0:IF X<1 THEN 150 ELSE Y
=INT(X):A=A+" ":GOSUB 200:A=A+B$
150 A=A+" Dollar":IF INT(X1)<>1 THEN A=A
+"s"
160 X=X-INT(X):II=X*100:A=A+" and ":IF I
I<10 THEN A=A+"0"
165 A=A+MID$(STR$(II),2)+"/100"
170 RETURN
200 IF Y<=10 THEN B$=A1(Y):RETURN
210 IF Y>10 AND Y<20 THEN B$=A2(Y-10):RE
TURN
220 B$=A3(INT(Y/10)):IF YMOD10<>0 THEN B
$=B$+"-"+A1(YMOD10)
230 RETURN
300 PRINT"Input amount: ";:LINEINPUTA:X=
VAL(A):IF X<.01 OR X>99999.99 THEN 300
310 GOSUB 100:PRINT A:GOTO 300
```

PCM



PRICKLY-PEAR SOFTWARE
QUALITY PROGRAMS FOR YOUR MODEL 100

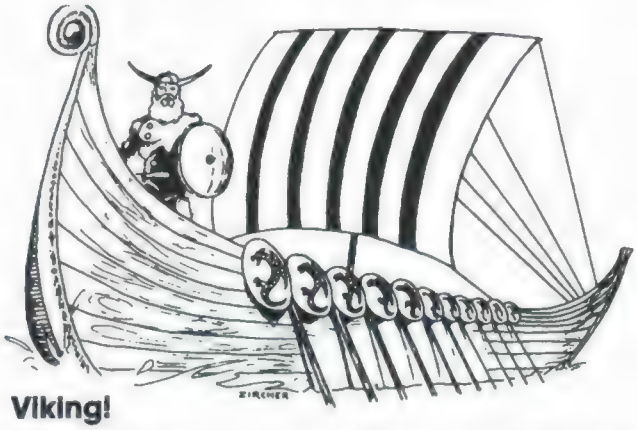


Monsters & Magic

The most realistic fantasy role-playing game yet for the Color Computer. You start out by rolling up your character's basic ability scores and buying equipment. When you are ready it's into the dungeon. Your character starts at first level, but can rise in levels by garnering experience in the fray. If you have role-played fantasy games you will be amazed at the realism of the combat system. Armor class, initiative, and damage by weapon type are all included, with over 50 different monsters to fight - each with it's own abilities. As you rise in level you can win treasure and find magic weapons and spells - if you live!! You set the game length by telling how many monsters you want to fight before you reach your final battle to the death against the powerful Dungeon Lord. There are 1000+ place description combinations in this text based game, and real excitement in every one! This is a fantasy simulation, and is truly not like any adventure game you have ever seen. For 1 player. **$19.95 tape—24K**

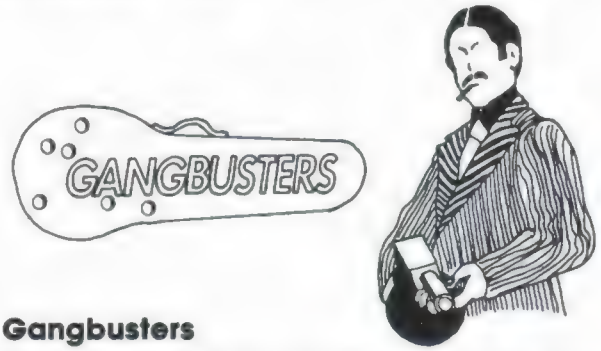
Astrology

Truly a classic, this program will accurately cast your complete horoscope. You just enter the date, time, and place of birth. The sun sign, rising sign, mid heaven (MC), lunar nodes, and planetary influences including houses and aspects between the planets will all be calculated, and a full chart drawn. You can also do progressed charts and transits. It will even tell you the day of the week you were born. The accompanying book will help you interpret this chart of your horoscope. The extent of the documentation is tremendous, even by our exceptionally high standards, and no previous knowledge of the subject is required. You can share in this wisdom which has been used for thousands of years in many cultures. This program was written by a professional Astrologer. **$34.95 tape — 16K**



Viking!

A simulation for 1 to 4 persons. Each begins as a landowner, and by farming their land, buying and selling land, expanding their fishing fleet, building on to their manufactory, increasing their population, equiping and training more soldiers, and regulating their taxes, each player tries to increase their economic power and rank until one becomes ruler over all. But beware plagues, rats, raiders, revolts, bad weather, and other misfortunes which may lie along the road to success. As you progress, see the map of your holdings increase. Playable in 1 to 2 hours, and different every time, you may have an addiction problem. **$19.95 tape — 24K**



Gangbusters

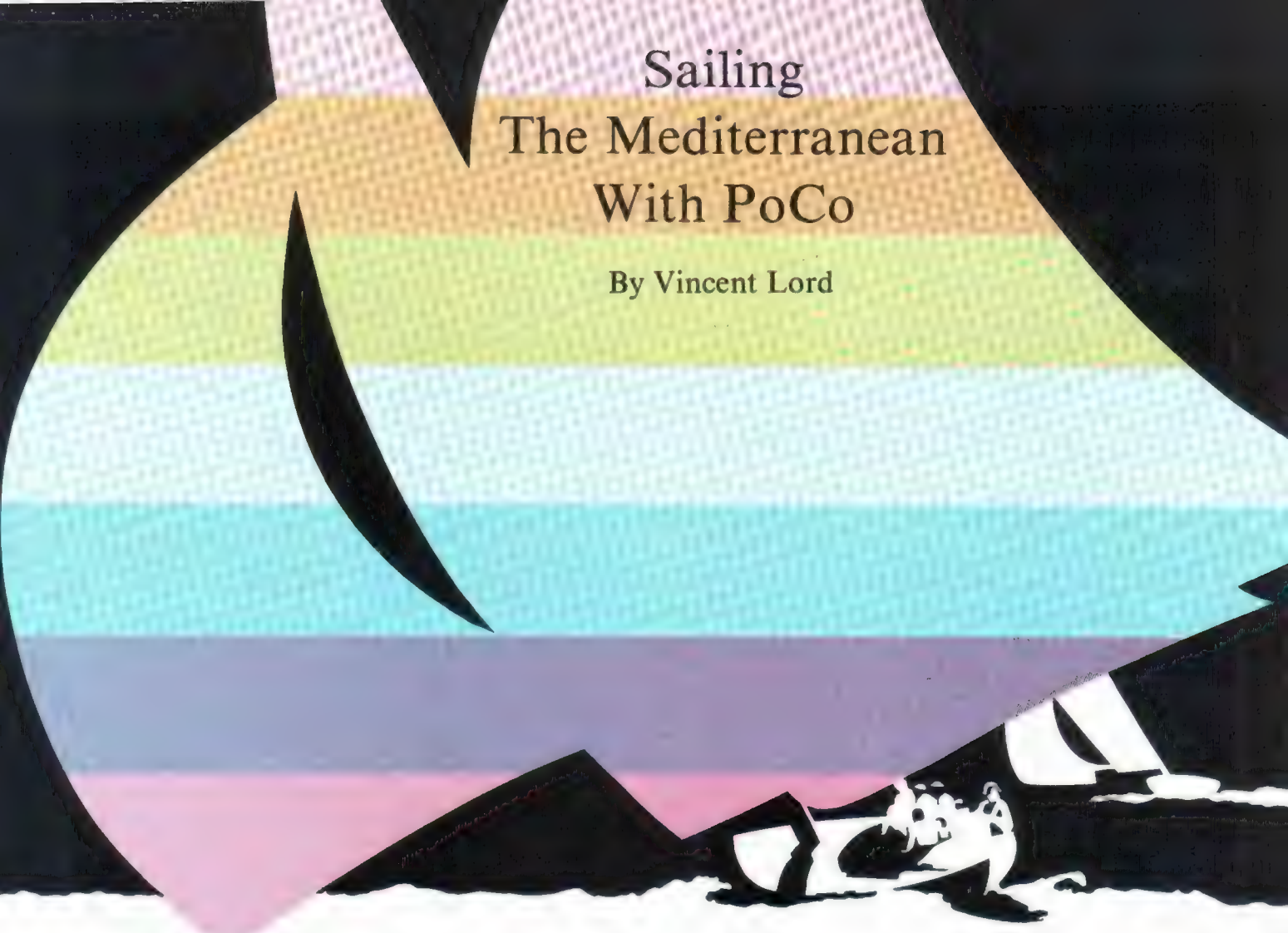
If you ever wanted to try a life of crime, this is your chance. You will start out as a Punk, but by using brains, and a little muscle, you can rise to become a Hood, Runner, Bookie, Torpedo, Fence, Kingpin, or win by becoming Syndicate Boss. Indulge yourself. Bribe a judge, or the District Attorney. Pay off the Cops. Take out a contract on another player, but watch out, they may be after you. Buy trucking companies, bootleg operations, houses of ill fame, but remember, if you get caught, you may do some hard time. Do you have what it takes to take over? This game will keep you close to your rod, get you thinking about bulletproof glass in your car, and definitely bring out the worst in you, but you'll love every minute of it. For 2 to 6 players, takes about 2 hours to play. Every game is excitingly different. **$19.95 tape — 24K**

Your personal check is welcome - no delay. Include $1.50 shipping for each program ordered. (Shipping free on $50.00 or larger orders). AZ residents add 7% sales tax. Orders shipped within two days.

Stocked by Quality Dealers, or

Send Order To: **PRICKLY-PEAR SOFTWARE**
9234 E. 30th Street
Tucson, Arizona 85710
(602) 886-1505

MasterCard VISA



Sailing The Mediterranean With PoCo

By Vincent Lord

Our world traveling publisher, Chief Petty Officer L. Falk, recently searched the world for the best use of the M100 in a realistic situation and displays his best efforts on the cover this month. If you happen to not recognize the scene, it's somewhere in the Mediterranean Sea, possibly near Greece? Only our publisher knows for sure. The pilot of his cruise ship obviously knows his way around for he has a computer to help and a CPO to program it for him. Unfortunately, the program was lost at sea and failed to be given proper publication. At the request of the CPO I will try to steer his ship home using our M100.

One of the most important items in navigation, whether by water or air, is a thorough understanding of the many factors that influence your ability to navigate. In airplanes there are primarily two factors; ie, airspeed and wind speed, the latter referred to as "currents." In boating or sailing there are several currents to consider. A definition of current refers to the horizontal movement of the water through which a vessel moves, and includes also wind, compass errors, steering inaccuracies, errors in engine RPM indications and ship performance (ie, fouled bottom, poor design, etc.).

Dead reckoning is a form of navigation used frequently because of its quick calculation (using a map and compass rose) and one finds by using dead reckoning that a vessel's course and speed are seldom if ever identical to the actual course or track made. The difference between the actual track and the intended track is caused by currents mentioned above.

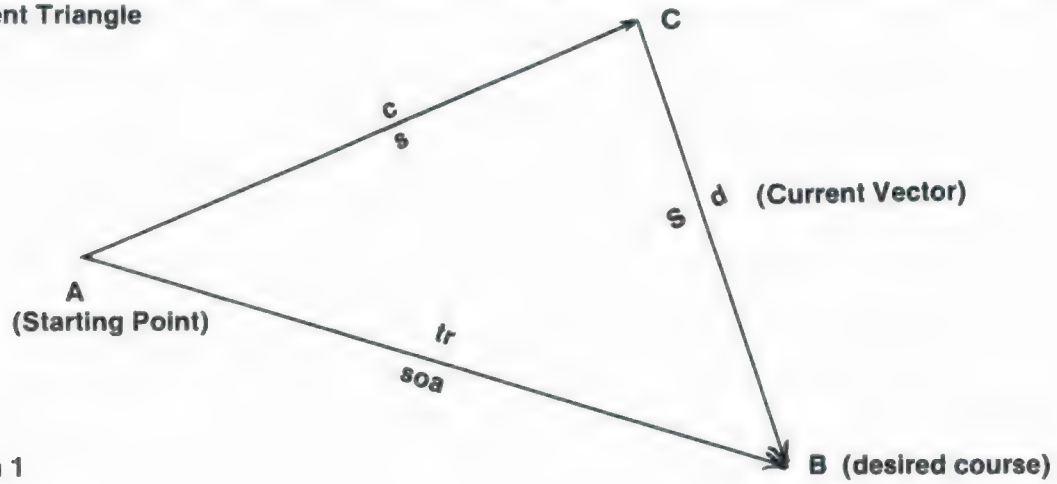
The more common method of determining the effects of currents involves using what is referred to as a "current triangle." The triangle in Figure 1 shows the six descriptions used in establishing the current triangle. The ordered course (c) and speed (s) are the direction and speed you would like to travel. The current set (S) and current drift (d) refer to the current direction and speed (example: wind or water current from 180 degrees at 6 knots, or from the south, blowing north, at 6 knots). The course

to follow to travel from point A to point B requires heading in a true heading direction (tr) that corrects for the current and at a resultant speed of advance (soa). A, B, and C refer to the angles used in the formulae.

From the current triangle (Figure 1) one can 1) find the resultant track (tr) and speed of advance (soa) when the vesels proceed at a given course (c) and speed (s)(OPTION 1 AT THE MENU), 2) find the course (c) and speed (s) to achieve the intended track (tr) and speed of advance (soa)(OPTION 2 AT THE MENU), or 3) find the current speed (s) and direction (d) when the two courses are known (OPTION 3 AT THE MENU). The computer program is self-instructing; however, remember that when you get current information such as winds, that means the direction FROM which the winds come. The other important consideration is the use of radians instead of degrees. The program automatically corrects for information entered in degrees. Degrees may be entered as decimal degrees, ie no minutes or seconds, and the resultant degrees are given as integers.

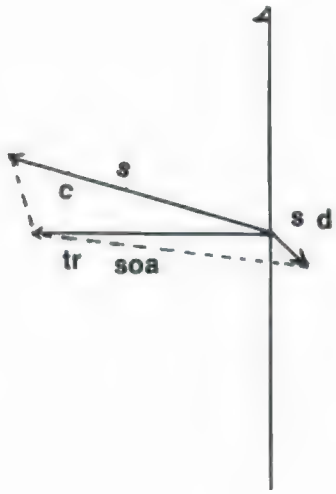
May you have smooth sailing.

Figure 1. Current Triangle



Menu Selection 1

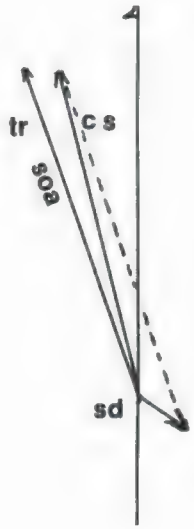
```
course direction(c)? 280
course speed(s)? 200
current direction(d)? 140
current speed(S)? 20

THE TRUE HEADING(tr) IS 276
DEGREES
THE SPEED OF ADVANCE (SOA)
IS 185.1 KNOTS

return to menu(1) or this section(2)
```



course as a result of current acting on true course.

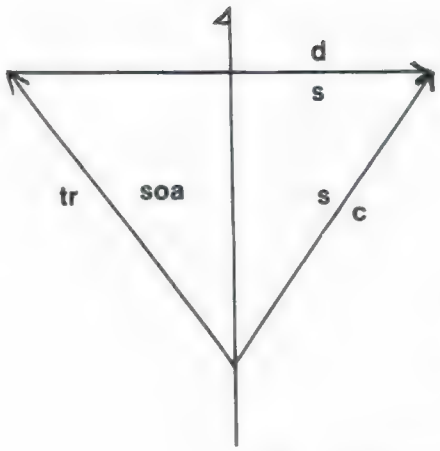
Menu Selection 2

```
this section determines the course and
speed to maintain soa and tr
true heading(tr)? 330
speed of advance(soa) 200
current heading(d)? 140
current speed(s)? 20

THE COURSE HEADING FOR tr
AND soa IS 334 DEGREES
THE SPEED FOR tr AND soa IS
185.1 KNOTS

return to menu(1) or this section(2)
```



course as a result of current acting on a course (to maintain true course)

```
THIS SECTION WILL FIGURE THE
CURRENT DIRECTION AND SPEED
true heading(tr)? 330
speed of advance(soa)? 200
course direction(c)? 40
course speed(s)? 200
THE CURRENT SPEED IS 229.4
  KNOTS
THE CURRENT DIRECTION IS 95
  DEGREES

RETURN TO MENU(1)
OR THIS SECTION(2)
```



Determination of current effects when true course and actual course are known

The listing:

```
5 CLS:DIMA(14),S(14)
10 LINE(100,12)-(120,36)
20 LINE-(100,36)
30 LINE-(100,12)
40 LINE(110,36)-(110,38)
50 LINE(85,38)-(123,38)
55 PRINT@175,CHR$(147)
60 LINE-(120,42)
70 LINE-(113,43)
80 LINE-(100,42)
90 LINE-(85,43)
100 LINE-(85,38)
110 LINE(0,44)-(239,48)
120 LINE(0,40)-(85,42)
130 LINE(0,55)-(239,60)
140 LINE(123,39)-(239,47)
150 PRINT@137,CHR$(146)
160 PRINT@3,"I'M POCO THE SAILOR MAN!"
165 PRINT@43,"I live in";TAB(20),"Prospe
ct Land!"
200 DATA 7456,20,6269,10,6269,20,6269,10
,7032,20,7456,20,6269,60
210 DATA 6269,20,5586,10,7032,20,5586,10
,4697,20,5586,20,6269,60
300 FOR X=1 TO 14
310 READ A(X),S(X)
320 SOUND A(X),S(X)
330 NEXT X
990 GOSUB 4900
1000 CLS:PRINT@18,"MENU"
1010 PRINT@43,"1) FIND TRACK (tr) AND SP
EED(SOA)"
1020 PRINT@83,"2) FIND COURSE (c) AND SP
EED(s)"
1030 PRINT@123,"3) FIND CURRENT (d) AND
SPEED(S)"
1040 PRINT@163,"4) END PROGRAM"
1050 INPUT"SELECTION (1-4)";Q
1060 ON Q GOSUB 1100,2000,3000,4000
1100 REM calculate tr and SOA
1105 CLS
1110 INPUT "course direction(c)";C
1120 INPUT"course speed(s)";SP
1130 INPUT"current direction(d)";D
1140 INPUT"current speed(S)";SC
1170 FX=SP*SIN(C*ATN(3D13)/90)+SC*SIN(D*
ATN(3D13)/90)
1180 FY=SP*COS(C*ATN(3D13)/90)+SC*COS(D*
ATN(3D13)/90)
1190 FR=SQR((FX*FX)+(FY*FY))
1200 AL=FX/FY
1210 ZL=ATN(AL)*90/(ATN(3D13))
1212 IF FX<0 AND FY>0 THEN ZL=360+ZL:GOT
01216
1213 IF FX<0 AND FY<0 THEN ZL=180+ZL:GOT
01216
1214 IF FX>0 AND FY<0 THEN ZL=180+ZL:GOT
01216
1215 IF FX>0 AND FY>0 THEN ZL=0+ZL
1216 CLS
1220 PRINT"THE TRUE HEADING(tr) IS";:PRI
NTUSING"####";ZL;:PRINT" DEGREES"
1230 PRINT"THE SPEED OF ADVANCE(SOA) IS"
;:PRINT USING"####.#";FR;:PRINT" KNOTS"
1250 PRINT@240,"return to menu(1) or thi
s section(2)"
1260 A$=INKEY$
1265 IF A$="" THEN 1260
1270 IF A$="1"THEN 1000 ELSE 1100
2000 REM detn heading and speed for true
 course
2005 CLS
2010 PRINT"this section determines the c
ourse and  speed to maintain soa and tr"
2015 INPUT"true heading(tr)";TH
2020 INPUT"speed of advance(soa)";SA
2030 INPUT"current heading(d)";CH
2040 INPUT"current speed(s)";CS
2050 IFTH<CH THEN HH=180+TH:GOTO 2090
2060 IFTH>CH THEN CH=180+CH:GOTO 2090
2070 IFTH=CH THEN GOTO 2500
2090 AB=HH-CH
2095 BA=AB*ATN(3D13)/90
2100 QW=SQR(SA*SA+CS*CS-(2*SA*CS*COS(BA)
))
2110 XC=(CS/QW)*SIN(BA)
2120 CX=ATN(XC/SQR(-XC*XC+1))
2125 FC=CX*90/ATN(3D13)
2130 NC=FC+TH
2135 CLS
```

```
2140 PRINT"THE COURSE HEADING FOR tr AND
 soa          IS ";:PRINT USING"###";NC;
:PRINT" DEGREES"
2150 PRINT"THE SPEED FOR tr AND soa IS";
:PRINT USING"####.#";QW;:PRINT" KNOTS"
2160 GOTO 2550
2500 REM headings in same direction
2510 PRINT"THE HEADING IS ";TH;" DEGREES
 AND THE SPEED IS ";SA+CS;" KNOTS"
2550 PRINT@240,"return to menu(1) or thi
s section(2)"
2560 A$=INKEY$
2570 IF A$="" THEN 2560
2580 IF A$="1" THEN 1000 ELSE 2000
3000 REM TO COMPUTE CURRENT AND SPEED
3010 CLS:PRINT"THIS SECTION WILL FIQUIRE
 THE CURRENT     DIRECTION AND SPEED"
3020 INPUT"true heading (tr)";JK
3030 INPUT"speed of advance(soa)";SO
3040 INPUT"course direction(C)";CD
3050 INPUT"course speed(S)";FS
3060 AG=JK-CD
3070 IF AG>180 THEN AG=360-AG
3075 IF AG>360 THEN AG=720-AG
3080 GA=AG*ATN(3D13)/90
3090 SS=SQR(FS*FS+SO*SO-(2*SO*FS*COS(GA)
))
3100 FF=(SO/SS)*SIN(GA)
3110 FH=ATN(FF/SQR(-FF*FF+1))
3120 FR=FH*90/ATN(3D13)
3130 IF CD>JK THEN DD=CD-FR ELSE DD=FR+C
D
3135 CLS
3140 PRINT"THE CURRENT SPEED IS ";:PRINT
USING"####.#";SS;:PRINT" KNOTS"
3150 PRINT"THE CURRENT DIRECTION IS ";:P
RINTUSING"###";DD;:PRINT" DEGREES"
3160 PRINT@240,"RETURN TO MENU(1) OR THI
S SECTION(2)"
3170 A$=INKEY$
3180 IF A$="" THEN 3170
3190 IF A$="1" THEN 1000 ELSE 3000
4000 MENU
4900 CLS:LINE(72,40)-(138,24)
4910 LINE-(132,48)
4920 LINE-(73,40)
5000 PRINT@170,"A"
5010 PRINT@104,"C"
5020 PRINT@263,"B"
5040 PRINT@137,"c"
5050 PRINT@178,"s"
5060 PRINT@181,"S"
5070 PRINT@218,"tr"
5080 PRINT@257,"SOA"
5090 PRINT@183,"d"
5100 FOR X=1 TO 5000:NEXTX
5110 RETURN
```

PCM

Complete Your TRS-80®
Genuine Radio S

8K Model 100
799.00
26-3801
As Low As $45 Per Month
On CitiLine Credit

24K Model 100
999.00
26-3802
As Low As $56 Per Month
On CitiLine Credit

A Powerful Workstation. The TRS-80 Model 100 is a true portable computer that works on batteries or with an AC adapter. It's small enough to fit easily in your briefcase, yet powerful enough to serve as your desktop microcomputer. And now Model 100 is complemented with a host of accessories for added versatility.

Ready to Use. The instant you turn on the Model 100, you can choose from one of five built-in management programs displayed on the eight-line by 40-character LCD display. With the full-size typewriter keyboard and built-in personal word processing program, you can jot down notes and write letters and reports. Model 100 also works as an appointment calendar, address book, phone directory and telephone auto-dialer. You can even write your own BASIC programs.

A TRS-80 Portable Terminal. With Model 100's communications program and built-in auto-dial modem, you can access national information services like CompuServe® and Dow Jones News/Retrieval®. Or connect Model 100 directly to another computer—micro, mini or mainframe—using the RS-232C interface. There are also parallel printer and cassette tape interfaces.

Available Nationwide. You can get the 8K Model 100 or the 24K Model 100, plus the accessories at your nearby Radio Shack Computer Center, participating store or dealer.

Direct-Connec

19.95
26-1410
Connects your Mod
phone jack for sendi
over phone lines. In
(non-prime time) on
Dow Jones News/R

Radio
The biggest name
A DIVISION OF TA

Prices apply at participating Radio Shack stores
CompuServe, Inc. Dow Jones News/Retrieval

Model 100 System with ack Accessories

Acoustic Coupler

39.95 26-3805

NEW!

Ideal for travelers who rely on their portable TRS-80 Model 100 for computer-to-computer communications or to access information services. Direct connection to a modular phone jack may be unavailable, so the Acoustic Coupler is your safety valve. You simply connect your Model 100 to a telephone handset via the Acoustic Coupler and you're ready to log on.

Modem Cable

100 to a modular and receiving data des one free hour n CompuServe and eval.

CCR-81 Cassette Recorder

59.95 26-1208

Save your program and data files on cassettes, or use Model 100's ready-to-run software. Includes cue/review, tape counter and LED data record indicator. Operates on AC or batteries (not incl.).

System Briefcase

49.95 26-3809

Components not included

NEW!

A lightweight and durable carrying case! System briefcase is spacious enough to hold your TRS-80 Model 100, the CCR-81 Computer Cassette Recorder, acoustic coupler and cables.

Shack®

n little computers®

DY CORPORATION

dealers. CompuServe is a registered trademark of gistered trademark of Dow Jones & Co., Inc

NEW 1984 TRS-80 COMPUTER CATALOG! Send for your free copy.

Mail To: Radio Shack, Dept. 84-A-726
300 One Tandy Center, Fort Worth, Texas 76102

NAME _____

COMPANY _____

ADDRESS _____

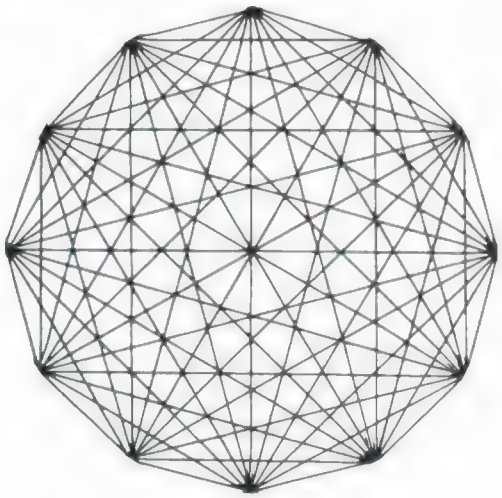
CITY _____ STATE _____ ZIP _____

TELEPHONE _____

LCD*

(*Little Computer Demo)

Here's a short but interesting program to demonstrate some of the neat things that can be done with the LCD (liquid crystal diode) display on the Model 100. The program asks the user to input a number of points. The response, which must be a number in the range of 4 through 32, dictates the number of sides of an equilateral polygon. The program then proceeds to connect each point with every other point. It makes an interesting display, especially with even numbers such as 8, 10, and 12. Inputting numbers higher than 15 will result in a picture that is more black than white, but it's still interesting to see how the lines are drawn. Inputting value of 32 will result in a completely darkened circle. Have fun!



The listing:

```
5 CLEAR 500:DEFSTR A:DEFINT I-N
10 CLS:PRINT"Input # of points: ";:LINEI
NPUT A
15 N=VAL(A):IF N=0 THEN MENU ELSE IF N<4
OR N>32 THEN 10
20 DIM X(33),Y(33)
25 R=30:Z=360/57.2958/N:CLS
30 FOR I=1 TO N:X(I)=R*COS(Z*(I-1)):Y(I)
=R*SIN(Z*(I-1)):NEXTI:X(N+1)=X(1):Y(N+1)
=Y(1)
145 FOR I=1 TO N:FOR J=I+1 TO N
160 X1=X(I)+120:Y1=32-Y(I):X2=X(J)+120:Y
2=32-Y(J)
165 LINE(X1,Y1)-(X2,Y2)
170 NEXT J:NEXT I
200 A=INPUT$(1):RUN
```

PCM

All work and no play makes Jack a dull boy . . .

NOW YOU CAN HAVE FUN WITH YOUR 24K MODEL 100

4 games for your enjoyment:

BLOCKADE — Your 'snake' grows longer as you hit the randomly appearing targets on the screen — as long as you don't run into yourself, a wall, or your opponent! A one or two player real-time game with optional sound and three speeds of play.

REVERSI — A 'board' game in which you try to outflank your opponent to end up with the most squares at game's end. Play against the computer or another human.

FRANKENSTEIN ADVENTURE — Find all of the necessary equipment to awaken the monster. The clues are hidden and the pitfalls are many!

ALEXIS ADVENTURE — Sail the seas and traverse the islands to recapture your kingdom. Chart your course or you may sail on forever!

SILVERWARE T.M.

— a lot of software for a little silver

PCM CERTIFICATION SEAL

All four games on tape for only $24.95
Calif. residents add 6% tax, overseas add $1.00 shipping

SilverWare PO Box 21101
Santa Barbara, CA 93121

Mastercard & Visa

Dealer inquiries invited

(805) 966-1449

Computer to Computer Communications With TELCOM

By Robert Frowenfeld

One of the most complicated tasks in realizing the full potential of compuers is the problem of having computer A talk to computer B. Sometimes it's just not enough to spend tens (if not hundreds) of hours developing a program that will make your computer stand on its head and spit wooden nickels—there's always somebody in the crowd who cries, "Hey, that's a great program, but can you transfer it to my computer?" Well, the TRS-80 Model 100 has what I believe to be one of the finest telecommunications packages available. And the cost can't be beat—it's free. Just power up your portable companion and you'll see a program called *TELCOM*.

Our friends in Fort Worth must have worked overtime on this one, because for me, a novice in computer-to-computer communications, *TELCOM* is the simplest thing in the world to operate. I own a TRS-80 Model 16 and a Model 100, and from time to time I've found *TELCOM* the solution to a lot of headaches. While my Model 100 is equipped with 24K of memory, there are times when the memory starts running low. I've found it very convenient to transfer my programs to my Model 16 and later, when I need them back, to load them back to the Model 100. I've used my old Model I's tape recorder to save and reload Model 100 programs, and it does an excellent job, but there's just no comparison to the speed and versatility of being able to save my Model 100 programs on an eight inch floppy. Imagine, being able to store over 400K of Model 100 programs and data on a single disk! I'm also in the process of converting a lot of my Model 16 programs to work on the Model 100. Having a 24-line by 80-column display available (on the Model 16) to perform the necessary modifications makes designing software for the Model 100 on the Model 16 a simple task indeed. By using the Model 100's *TELCOM* program, the Model 16's TERMINAL package (which, incidentally, also comes free with the Model II, 12, and 16), and an RS-232 serial communications cable, transferring programs back and forth between the two machines has become a breeze. Oh yes, lest I forget—an additional advantage to being able to load programs from the Model 16 to the Model 100 is . . . (drum roll please) . . . SCRIPSIT. Using the Model 100's built in word processor, it's great to be able to create documents on the Model 100, load them into Model 16 SCRIPSIT, make the necessary formatting changes, spelling corrections, etc., and then print out the final product. For you executives out there in PCM land—this could be the solution to a lot of problems!

But now, let's get down to the nuts and bolts involved in getting the Model 100 (and *TELCOM*) and the Model II, 12, or 16 (and TERMINAL) to talk to each other. First, you must have the proper cable connecting the two computers together. Radio Shack has a Model 100 serial communications cable available (part #26-4403), and a null modem adapter (part #26-1496). You'll need both of these because the computers aren't talking through a modem;are talking directly to each other. For those of you who like to get technical, the null modem adapter switches cable line pairs 2,3 and 4,5 as well as rearranges lines 6,8, and 20 so the two computers can communicate with each other. If you find the desire to save the $40 it will cost to purchase the cable and null modem adapter, you can build your own for just a few dollars by following the diagram in the Model II, 12, or 16 owner's manual in the Technical Section under the heading of Model II to Model II communications. Its very easy to follow (If I can do it, anyone can.).

So, now you have your cable and null modem adapter plugged together with one end of the cable plugged into the Model 100. Take the other end and plug it into Serial Channel A of your Model 16. So much for the mechanics.

From TRSDOS Ready on the Model 16 type TERMINAL (press ENTER) and place the cursor on the Model 100's menu over TELCOM and press ENTER. Would

you believe we are less than two minutes away from getting the two computers talking to each other?

On the Model II, type the letter "S" and press ENTER, and on the Model 100, press the function key F3. We are now ready to enter the communications protocols for each computer. For each machine, this will consist of transfer (Baud) rate, word length, parity, and stop bits. On the Model II, TERMINAL will display: Enter TRSDOS Command (1-79), indicating that you should enter a command line between 1 and 79 characters in length. You must now type: SETCOM A=(300,7,E,1) and press ENTER. The Model II will respond with: PORT A ENABLED AS (-BAUD = 300, BITS = 7, PARITY = E, STOPS =1. The Model 100 will be prompting indicating that it is awaiting

"Our friends in Fort Worth must have worked overtime on this one, because for me, a novice in computer-to-computer communications, *TELCOM* is the simplest thing in the world to operate. I own a TRS-80 Model 16 and a Model 100, and from time to time I've found *TELCOM* the solution to a lot of headaches."

a command to change the communications protocol. Simply type: 37E1D. The two computers are now set up to "talk to each other" using the settings: Baud Rate = 300, Word Bits = 7, Parity = you with the word "Stat," Even, and Stop Bits = 1. It really doesn't matter too much *what* the settings are; what really matters is that they must be *identical* for both machines.

Now for the last step. On the Model II, type the letter "T" and press ENTER; the screen will go blank. On the Model 100, press the function key F4. That's it! If you did everything correctly, anything you type on one computer's keyboard will be echoed on the other. If you find that you can type on the Model 100 and, although it echos on the Model II it is not displaying on the Model 100, press the function key F4 on the Model 100 to change the duplex mode from Full to Half. Similarly, if you are typing on the Model II and you can't see what you are typing, you will have to get back to the Communications Menu (by pressing BREAK) and issuing the command "E" to toggle the self-echo option.

Now that the Model 100 and Model II are talking to each other, transferring whole files is a breeze. The one thing to keep in mind if you are transferring a BASIC program is that it must be in ASCII format. To save a file in this format on either machine, use the "A" option when saving the program. As an example, to save the program named XYZ, you would type: SAVE "XYZ",A.

To upload a program from the Model 100 to the Model II, the steps a—j must be performed in order:

a) Model 100: Select the file to upload and make sure it is in ASCII format.
b) Model II: Call up the TERMINAL program, set up the communications protocol.
c) Press "R" and hit ENTER.
d) Press "Y" to open the RAM buffer.
e) Press "Y" to reset the RAM buffer (if necessary).
f) Model 100: Enter *TELCOM*, Press function key F4 to get into TERM mode.
g) Press function key F3 to select the UPLOAD option.
h) In response to the "File to Upload?" question, enter the file name and extension to transfer.
i) Model II: Press "T" to enter the TERMINAL mode.
j) Model 100: In response to the "Width:" question, hit ENTER.
k) Model II: Press BREAK key to return to TERMINAL menu.
l) Press "C" to copy RAM buffer to disk.
m) Respond to "Enter Filespec?" with file name to save on Model II.

If you've done everything correctly, you should see the program being transferred on the video display of the Model II. Depending on the setting on the duplex on the Model 100, you may (or may not) also see the file as it is sent. Follow steps k—m above to download the Model II's RAM buffer to disk.

To transfer files in the other direction from the Model II to the Model 100, follow the procedure outlined below:

a) Model II: Select the file to upload and make sure it is in ASCII format.
b) Call up the TERMINAL program, set up the communications protocol.
c) Press "G" and hit ENTER to copy file into RAM buffer.
d) In response to "Enter Filespec?" enter file name to transfer.
e) Model 100: Enter *TELCOM*, Press function key F4 to get into TERM mode.
f) Press function key F4 to select the DOWNLOAD option.
g) In response to the "File to Download?" enter the file name and extension to transfer.
h) Model II: Press "X" and hit ENTER to enter TERMINAL and TRANSMIT mode.
i) After transmitting, press BREAK to return to main menu.

A few notes of caution and assistance. First, to eliminate the thing that results in problems more often than anything else, make sure that the two machines have the same communications settings: Baud rate, parity, word length and stop bits. Second, make sure the RAM buffer is clear on the Model II before you transfer anything into it. This can be done by entering a "D" (for display) from the TERMINAL main menu. Also, make sure your connections are good and tight, and don't *ever, ever, ever,* turn off either machine or pull out any cables until you have *turned off* the communications function. On the Model II, this can be done with the command SETCOM A=OFF from TRSDOS or TERMINAL. On the Model 100, *TELCOM* can be exited by pressing the function key F8. If, by accident, you don't turn off the channels, a "glitch" in the line (which can be caused by pulling out a cable, turning off the machine, or letting your Model 100 shut itself off if it remains unused for a few minutes) can make either (or both) machines lock up.

So there you have it. It's really not as difficult as it sounds. Give it a try and see if you can be the first person on your block to get your Model 100 and Model II, 12 or 16 talking to each other. Happy communicating!

PCM

So you want that program out there? Don't just download it, download it with . . .

DOWNLOAD AID

By Guerri F. Stevens

Did you ever download a BASIC program from CompuServe and find that it had extraneous carriage returns? Have you transmitted a Model II program to your Model 100 and discovered that the control Js you used were causing DS errors? If either of these things has happened to you, the program which follows can help.

The *Download Aid* program analyzes a BASIC program stored in ASCII. It replaces the sequence of characters produced by the control J on the Model II with blanks. It attempts to remove extraneous carriage return characters. I say "attempts" because this feature is not (and cannot be) foolproof, as I will explain later.

Operating Instructions

The program runs on the Model 100 and processes a BASIC program stored in ASCII (.DO extension). At the beginning, the *Aid* program clears the screen and displays the title *Download Aid*. It then asks you for certain information. Type the appropriate responses, as described below. If you make an error, the message "ERROR — RETRY" is displayed in reverse video and the request is repeated.

File Name:

Type the name of the file you want to process. Remember that the file must be a BASIC program stored in ASCII form.

If the file is on cassette, you must precede the name with "CAS:". If you don't, the program assumes that the file is in RAM.

If the file is on cassette, put the cassette into your cassette recorder, and press PLAY.

Output on <C>assette or in <R>AM?

If your file is on cassette, you will not get this message; the output file will be placed in RAM automatically.

If your file is in RAM, you may choose the output device; if you want the output file to be written on cassette, type C (upper or lower case is acceptable). If you want the output file in RAM, type R.

If you want the file on cassette, put a cassette into your recorder and set it to record. This is usually done by depressing the PLAY and RECORD buttons simultaneously.

Output file name is F@@xxx

No response is necessary. This message tells you what the output file will be called. The name is always F@@ followed by the first three characters of the input file name.

File F@@xxx already exists.
Do you want to kill it?

Your output file is going to be in RAM but a file of the same name already exists. If you want to write over it, thus destroying (killing) the file, type "yes." If you want to save the existing file, type "no" and the *Aid* program will end.

If your output file will be on cassette, the program cannot check to see if a file of the same name exists. It is up to you to insert the appropriate cassette in your recorder.

At this point, the *Aid* program starts to process your file. When it is done, it will ask you to rewind the cassette if either your input or output file was on cassette:

Set recorder to REWIND; press ENTER?

Set your recorder to rewind and press the ENTER key on the Model 100. The cassette will begin to rewind.

Press ENTER when rewinding is done?

When the cassette has been completely rewound, press the ENTER key again.

The word OK tells you that the *Aid* program is done. Load the output file (F@@xxx) and make sure it is correct.

Extraneous Carriage Returns

The *Aid* program detects an extraneous return by noticing that a BASIC line begins with a non-numeric character. This technique will not identify an extra carriage return if the return precedes a number such as the line number after GOTO. After the *Aid* program has finished, load the output file from BASIC. Any extra carriage returns which precede numbers will be flagged with "UL" errors. If you get any, list the program while in BASIC. The last line listed is the one containing the error. Use *TEXT* to delete the carriage return, and try again.

Program Description

The program is listed below. Each section is preceded by a brief comment. The first section clears string space and controls the flow of logic through the other sections.

The second section displays the title and gets the name of the file to be processed. It performs another section, if necessary, to get the output device. It

then displays the name of the output file. If the output device is RAM, it calls a routine to check whether the output file already exists. It then opens the input and output files.

The section entitled "Process a Line" reads the input file one line at a time. Each line is checked for the Model II's control J: a string of the three hexadecimal characters 0A 0D 0A. If found, they are replaced by blanks. To handle excess carriage returns, the line's first character is checked to see whether it is numeric. If not, it is assumed to be a continuation of the prior line. If an extra carriage return happens to precede a number (such as the line number following a GOTO), this technique will not identify it as an extra carriage return.

The last two sections rewind the cassette when necessary, and display the error message.

The variables used in the program are listed below. Please note that all variables beginning with "I" are integers and all those beginning with "S" are strings.

I1 General-purpose integer variable
IS Switch for testing existence of a RAM file with the same name as the output file
SF Source file name
SG General-purpose string variable
SL Program line being processed
SN Source file name without device prefix
SO Output file name
SP Prior program line
ST File type code:
- C = cassette input, RAM output
- R = RAM input, RAM output
- T = RAM input, cassette output
- X = RAM input, output not yet known

SV 3-character string representing Model II control J

The listing:

```
 10 REM ************************************
*****************
20 REM * REMOVE EMBEDDED LINE FEEDS FROM
 BASIC PROGRAM *
30 REM * WRITTEN BY GUERRI F. STEVENS
                *
40 REM *         GENERAL BUSINESS SYSTEMS
, INC.          *
50 REM ************************************
*****************
60 CLEAR 1000
70 MAXFILES=2
80 DEFSTR S
90 DEFINT I
100 GOSUB 210
110 IF IS>0 THEN END
120 IF NOT EOF(1) THEN GOSUB 1010: GOTO
120
130 IF SP<>"" THEN PRINT #2,SP
140 CLOSE
150 IF ST<>"R" THEN GOSUB 1130
160 END
```

```
170 'SKIP3
180 REM **************
190 REM * INITIALIZE *
200 REM **************
210 CLS
220 PRINT TAB(14);"Download Aid"
230 PRINT
240 SV=CHR$(10)+CHR$(13)+CHR$(10)
250 SF=""
260 IF SF="" THEN GOSUB 410: GOTO 260
270 SO=""
280 IF ST="X" THEN GOSUB 560: GOTO 280
290 SO=SOP"F@@"
300 IF LEN(SN)<4 THEN SO=SO+SN ELSE SO=S
O+LEFT$(SN,3)
310 PRINT "Output file name is ";SO;
320 IF ST="T" THEN PRINT ELSE PRINT ".DO
" : GOSUB 660
330 IF IS>0 THEN RETURN
340 OPEN SF FOR INPUT AS 1
350 OPEN SO FOR OUTPUT AS 2
360 RETURN
370 'SKIP3
380 REM *****************
390 REM * GET FILE NAME *
400 REM *****************
410 ST="X"
420 LINEINPUT "File Name: ";SF
430 IF SF="" THEN GOSUB 1220: RETURN
440 I1=G(SF,":")
450 IF I1=0 THEN SN=SF : RETURN
460 IF I1<>4 OR LEN(SF)=4 THEN SF="" : G
OSUB 1220: RETURN
470 SG=LEFT$(SF,I1-1)
480 SN=RIGHT$(SF,LEN(SF)-4)
490 IF SG="RAM" OR SG="ram" THEN RETURN
500 IF SG="CAS" OR SG="cas" THEN ST="C"
ELSE SF="" : GOSUB 1220
510 RETURN
520 'SKIP3
530 REM ******************************
540 REM * ASK ABOUT CASSETTE OUTPUT *
550 REM ******************************
560 SG=""
570 INPUT "Output on <C>assette or in <R
>AM";SG
580 IF SG]"R" OR SG="r" THEN ST="R" : RE
TURN
590 IF SG="C" OR SG]"c" THEN ST="T" : SO
="CAS:" : RETURN
600 GOSUB 1220
610 RETURN
620 'SKIP3
630 REM ******************************
640 REM * TEST EXISTING OUTPUT FILE *
650 REM ******************************
660 ON ERROR GOTO 840
670 IS=1
680 OPEN SO FOR INPUT AS 1
690 ON ERROR GOTO 0
700 IF IS=0 THEN RETURN
710 CLOSE
720 PRINT
730 PRINT "File ";SO;" already exists."
740 SG=""
750 IF SG]"" THEN GOSUB 900: GOTO 750
760 IF SG="NO" THEN PRINT CHR$(27);"pJob
 terminated";CHR$(27);"q" :  RETURN
770 IS]0
780 KILL SO+".DO"
790 RETURN
800 'SKIP3
810 REM ****************************
820 REM * NON-EXISTENT FILE IS OK *
830 REM ****************************
840 IF ERR=52 THEN IS=0 : RESUME NEXT
850 ON ERROR GOTO 0
860 'SKIP3
870 REM ******************
880 REM * KILL OLD FILE? *
890 REM ******************
900 INPUT "Do you want to kill it";SG
910 IF SG="yes" THEN SG="YES" : RETURN
920 IF SG="no" THEN SG="NO" : RETURN
930 IF SG="YES" OR SG="NO" THEN RETURN
940 SG]""
950 GOSUB 1220
960 RETURN
970 'SKIP3
980 REM ******************
990 REM * PROCESS A LINE *
1000 REM ******************
1010 LINEINPUT #1,SL
1020 IF LEN(SL)=0 THEN RETURN
1030 I1=INSTR(SL,SV) : IF I1>0 THEN MID$
(SL,I1)="    " : GOTO 1030
1040 I1=ASC(LEFT$(SL,1))-48
1050 IF I1<0 OR I1>9 THEN SP=SP+SL : RET
URN
1060 IF SP<>"" THEN PRINT #2,SP
1070 SP=SL
1080 RETURN
1090 'SKIP3
1100 REM ******************
1110 REM * REWIND CASSETTE *
1120 REM ******************
1130 INPUT "Set recorder to REWIND; pres
s ENTER";SG
1140 MOTOR ON
1150 INPUT "Press ENTER when rewinding i
s done";SG
1160 MOTOR OFF
1170 RETURN
1180 'SKIP3
1190 REM ************************
1200 REM * DISPLAY ERROR MESSAGE *
1210 REM ************************
1220 PRINT TAB(5);CHR$(27);"p***** ERROR
 - RETRY *****";CHR$(27);"q"
1230 RETURN
```

PCM

Let's Go Spelunking But Watch Out For The Wumpus

by Vincent Lord

Wumpus is a nice little creature who just loves to take you on a spelunking adventure through a maze of 36 caves all interconnected by means of three passageways from each cave to three more caves. You must crawl through mud, dust, water, and total darkness to travel from one cave to the next. Your senses will guide you and forewarn you. For instance, if you feel a draft, then watch out, for you are one cave away from a bottomless pit which will swallow you up, and we will never hear from you again. If your hearing is good, then you'll recognize the high pitch chirping sound as coming from bats one cave away. The bats will become groupies with enough power to lift you up and transfer you to another section of the caves. Watch where you are going. It's best to avoid these pitfalls, but sometimes you just can't.

The object, of course, is to meet our travel host on your terms. He is rather obnoxious in both smell (you can smell him up to two caves away) and behavior. Try not to stumble across his path, wake him up, or, when you think you have him in sight, shoot at him and miss.

Now for the details. After typing RUN, wait for several minutes for the random number generator to fill the matrices (lines 9990—9996). You will see a notice of which cave you are in and the three caves joining. Keep track of this information. To move one cave away (to one of the three caves listed) simply type in the adjoining cave number (1 to 36). You can move up to three caves by typing the cave numbers separated by commas. When you think you have finally located our stinky friend, you may shoot at him by typing "S" and the cave number(s)—up to three away. If you get him, well, what can I say but your next adventure should be as a guide in the backwoods of New York City; however, if you miss, then it's rough sailing. You may never find yourself as the wumpus carries you through the caves upsetting all the creatures living there and causing them to move on. The program was adapted for the M100 from an original program run on the DEC10 and created by my good friend and gamesman George Ellsworth in Attica, Indiana. My thanks to him for the basic program structure and permission to modify his program.

The listing:

```
500 CLEAR 300
510 DIMC(36,3),C1(36,3),R(36),B$(36),P$(
36),W$(36),I(6)
515 AR=5
520 FOR X=1 TO36:FOR Y=0TO3:READC(X,Y):N
EXTY,X
530 FOR X=1TO36
540 GOSUB9990:A=INT(37*DU):IFR(A)=0 THEN
R(A)=X ELSE540
550 NEXTX
560 FOR X=1TO36:FORY=0TO3
570 C1(X,Y)=R(C(X,Y)):NEXTY,X
576 FORX=1TO36:P=C1(X,0)
578 FOR Y=0TO3:C(P,Y)=C1(X,Y)
580 NEXTY,X
590 FORX=1TO5
600 GOSUB9990:B=R(INT(31*DU)):IFB$(B)>""
THEN600
610 B$(B)="BATS"
620 GOSUB9990:P=R(INT(31*DU)):IF P$(P)>"
"THEN 620
630 P$(P)="PIT":NEXT X
640 GOSUB9990:W=INT(7*DU):W$(R(W))="WUMP
US"
650 FORXP=1 TO 40:GOSUB9990:L=R(INT(7*DU
```

```
)+30)
660 IF P$(L)+B$(L)+W$(L)>""THEN 690
670 GOSUB 1500
680 IF P$+W$+B$=""THEN700
690 NEXT XP:GOTO500
700 CLS:PRINT"you are in cave # ";L
710 PRINT"You have passageways to:"
720 FORX=1TO3:PRINTC(L,X);:NEXT X
730 GOSUB 800:GOSUB 1500
750 PRINTB$:PRINTP$:PRINT W$
760 GOTO1000
800 IF T=0 THEN RETURN
810 PRINT"The WUMPUS is awake!!"
820 FORX=1 TO 36:IF W$(X)="WUMPUS" THEN
W$(X)="":GOTO 840
830 NEXT X:STOP
840 GOSUB9990:M=INT(4*DU):X1=C(X,M):IF X
1=L THEN CLS:GOTO880
850 IF C(X1,M)<>X THEN X=C(X1,M) ELSE X=
X1
870 IF X<>L THEN 890
875 CLS
880 PRINT"THE Wumpus FOUND YOU AND":PRIN
T"HAD YOU FOR LUNCH":GOTO 9050
890 W$(X)="WUMPUS":GOSUB9990:IFINT(11*DU
)>6 THEN T=0
900 IF AR>0 AND T=0 THEN PRINT"THE Wumpu
s HAS GONE BACK TO SLEEP"
905 IF AR>0 THEN PRINT"YOU HAVE"AR"ARROW
S LEFT":RETURN
910 PRINT"AND YOU'RE OUT OF ARROWS. THE
Wumpus IS  AWAKE, AND YOU ARE HELPLESS.
IT'S  ONLY A MATTER OF TIME..."
920 GOTO 880
1000 H$="":FORZ=1TO6:I(Z)=0:NEXTZ:Z=0:GT
=0
1010 PRINT"PLOT YOUR STRATEGY:"
1020 A$=INKEY$:IF A$=""THEN GOTO 1020
1030 IF A$<>CHR$(29) THEN PRINTA$;
1040 IF A$="S" AND GT=0 THEN GT=2:GOTO 1
020
1050 IF A$="M" AND GT=0 THEN GT=1:GOTO 1
020
1070 IF A$=CHR$(13) THEN 1120
1080 IF ASC(A$)>47 AND ASC(A$)<58 THEN 1
110
1090 IF ASC(A$)=32 OR ASC(A$)=44 THEN IF
 H$>"" THEN 1120 ELSE 1020
1100 PRINT"BAD INPUT--TRY AGAIN":PRINT"
 ":GOTO 1000
1110 H$=H$+A$:IF GT=0 THEN GT=1
1115 GOTO 1020
1120 Z=Z+1:I(Z)=VAL(H$):H$="":IF Z<>5 AN
D A$<>CHR$(13) THEN 1020
1200 IF I(1)=777 THEN 2000
1205 ON GT GOTO 1210,1350
1210 FORX=1TO6:IFI(X)=0 THEN 700
1220 FOR C=1TO3:IF C(L,C)=I(X) THEN L=I(
X):GOSUB 1250:GOTO1240
1230 NEXTC:CLS
1235 PRINT"YOU'RE LOST!  YOU CAN'T GET":
PRINT"THERE FROM HERE.  NOW YOU'RE":PRIN
T"WANDERING THRU CAVES:":GOTO1600
1240 NEXT X
1250 IF W$(L)="WUMPUS" THEN CLS:PRINT"YO
U FOUND THE Wumpus AND HE":PRINT"ATE YOU
 FOR BREAKFAST!":GOTO 9080
1260 IF P$(L)="PIT" THEN CLS:PRINT"YOU F
ELL INTO A BOTTOMLESS PIT!":PRINT" BYE..
.";:GOTO 9000
1270 IF B$(L)<>"BATS"THEN RETURN
1280 CLS:PRINT" A GIANT BAT HAS YOU ";:P
RINT"      IN HIS GRASP!     ";
1295 B$(L)=""
1300 GOSUB9990:B=INT(37*DU):IF B$(B)="BA
TS" THEN 1300 ELSE B$(B)="BATS"
1305 GOSUB9990:L=INT(37*DU): IF B$(L)="B
ATS" THEN 1305
1310 PRINT"HE DROPPED YOU IN CAVE"L;
1315 FOR J=1 TO 1000:NEXT J
1320 GOSUB1250:GOTO700
1350 LA=L:AR=AR-1:T=1
1360 FORX=1TO6:IF I(X)>0 THEN 1370 ELSE
CLS
1365 PRINT"YOU MISSED HIM":GOTO700
1370 IF X<5 THEN IF I(X)=I(X+2) THEN CLS
:PRINT"ARROW DAMAGED BY FULL REVERSE":PR
INT"AT CAVE #"I(X+1)"! DEVICE NOW BACK":
PRINT"TRACKING!  LOOKOUT...":PRINT:LA=L:
GOTO1415
1380 FOR C=1TO3:IF C(LA,C)=I(X) THEN LA=
I(X):GOTO1430
1390 NEXTC:CLS:PRINT"YOU PLOTTED AN IMPO
SSIBLE"
1395 PRINT"ROUTE WITH YOUR ARROW. IT IS"
1400 PRINT"NOW RUNNING AMUCK THRU CAVES"
1405 FORC=X TO 5: GOSUB9990:X=INT(4*DU):
 LA=C(LA,X):PRINT LA;
1408 FOR Q=1 TO 800:NEXT Q
1410 IF W$(LA)="WUMPUS" THEN CLS:PRINT"T
HE WUMPUS WAS IN CAVE"LA,"YOU GOT HIM!":
GOTO 8000
1413 IF LA=L THEN CLS ELSE 1420
1415 PRINT"YOUR ARROW RETURNED TO YOUR":
PRINT"CAVE AND SHOT YOU! (OOPS...)":GOTO
9000
1420 NEXT C:GOTO1365
1430 IF W$(LA)="WUMPUS" THEN 1410
1435 IF LA=L THEN 1413
1440 NEXT X:STOP
1500 P$="":B$="":W$=""
1505 IF AR=0 THEN 910
1510 FOR X=1TO3
1520 IF B$(C(L,X))="BATS"THENB$="I HEAR
BATS"
1530 IF P$(C(L,X))="PIT"THENP$="I FEEL A
 DRAFT"
1540 IF W$(C(L,X))="WUMPUS" THEN W$="I S
MELL THE WUMPUS"
1550 Y=C(L,X):FOR Z=1TO3
```

```
1560 IF W$(C(Y,Z))="WUMPUS" THEN W$="I S
MELL THE WUMPUS"
1570 NEXT Z,X
1580 RETURN
1600 FOR X=1TO5
1610 GOSUB9990:L=C(L,INT(4*DU)):PRINT L;
1620 FOR B=1 TO 1500:NEXT B:GOSUB 1250
1630 NEXT X
1640 GOTO700
2000 CLS:CK=0:PO=0:FORX=1 TO 36
2010 PRINT@PO,CHR$(138);X;R(X);:PO=PO+8:
CK=CK+R(X)
2020 NEXTX:PRINTCK:CK=0
2023 FORX=1TO36:FORY=0TO3:CK=CK+C(X,Y)
2025 NEXTY,X:PRINTCK,666*4
2030 IF INKEY$="" THEN 2030
2035 Y=1:Z=12
2040 CLS:FORX=Y TO Z
2050 PRINTX;B$(X);P$(X);W$(X)
2060 NEXTX
2065 IF INKEY$="" THEN 2065
2070 Z=Z+12:Y=Y+12:IF Z>36 THEN 700
2080 GOTO 2040
5000 DATA 1,2,6,9,2,1,11,3,3,2,13,4,4,3,
15,5,5,4,17,6,6,5,7,1,7,8,6,18,8,9,21,7,
9,8,1,10
5010 DATA 10,9,23,11,11,10,2,12,12,11,25
,13,13,12,3,14,14,13,27,15,15,14,4,16,16
,15,29,17,17,16,5,18
5040 DATA 18,17,19,7,19,30,18,20,20,19,3
2,21,21,20,8,22,22,21,33,23,23,22,10,24,
24,23,34,25,25,24,12,26
5060 DATA 26,25,35,27,27,26,14,28,28,27,
36,29,29,28,16,30,30,29,31,19,31,30,32,3
6,32,31,20,33,33,32,22,34
5080 DATA 34,33,24,35,35,34,26,36,36,35,
28,31
8000 PRINT"CONGRADULATIONS"
8010 FOR P=1 TO 1000:NEXT P
8900 MENU
9000 REM PIT OR SHOT SELF
9010 PS=PS+1
9020 ON PS GOTO 9025,9030,9035
9025 PRINT"THAT'S ONE...":GOTO 9040
9030 PRINT"THAT'S TWO...":GOTO 9040
9035 PRINT"THAT'S THREE!!!":GOTO 9050
9040 FOR X=1 TO 1000:NEXTX:GOTO650
9050 PRINT"BEING EATEN IS FINAL":FOR X=1
 TO 1200:NEXTX
9080 CLS:PRINT"GAME OVER";:FOR X=1TO 120
0:NEXTX
9090 PRINT"  ":PRINT"  "
9095 MENU
9990 REM random number generator
9991 SEC=VAL(RIGHT$(TIME$,2))
9992 FOR I=1TOSEC STEP 9
9993 DUMMY=RND(1)
9994 NEXT I
9995 IF DUMMY<1/37 THEN 9990
9996 RETURN
```

Prevent damage to your cassette plugs with this . . .

Homemade PoCo To CCR Cable

By William T. Stauffer

Radio Shack has done an outstanding job in the design of the PoCo and its peripherals. The System Briefcase has been designed in such a manner as to permit operation of the system (PoCo and cassette tape recorder) without removing them from the briefcase; however, there is one drawback—when the cassette recorder cable is plugged into the cassette recorder, the plugs extend to the outside of the briefcase and could cause damage to the case if it is closed with the cassette recorder connected. The obvious solution to this problem is the use of low profile plugs which, to my knowledge, are not available. Since the required plugs are not readily available, an alternative solution is to make them—in fact, why not combine the three plugs into one unit—a "tri-plug"?

Following this line of thinking, and with a close study of the Radio Shack catalog, I devised and built a new PoCo-to-CCR cable. The details of the fabrication of the cable, with referenced Radio Shack catalog part numbers, follow:

1) Prepare two ⅛" miniature phone plugs (274-286) and one $^{3}/_{32}$" submini phone plug (274-289) by removing the cover, bending off the ground lug (the long lug connected to the threaded portion of the plug), and cutting off the center lug to leave approximately ⅛".

2) Insert the miniature plugs into the "EAR" and "AUX" jacks of the CCR-81 and the submini plug into the "REM" jack.

3) Cut a piece of printed circuit board (276-1586) 2" x ¾", Figure 1.

4) Scribe a center line on the printed circuit board.

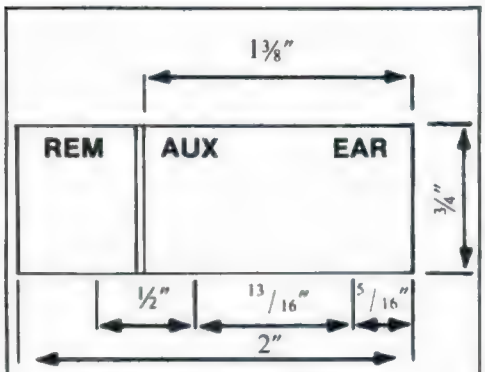


Figure 1.

Printed Circuit Board Layout

5) Locate hole centers corresponding to the centers of the "EAR," "AUX," and "REM" jacks of the CCR-81 on the center line. These dimensions are ½" between the "REM" and "AUX" and $^{13}/_{16}$" between the "AUX" and "EAR."

NOTE: These dimensions are for the Radio Shack CCR-81. If you are using the cable with another type of cassette tape recorder such as the Radio Shack CTR-80, check the hole centers and drill your board accordingly.

6) Drill ⅜" holes for the "AUX" and "EAR" plugs and ¼" hole for the "REM" jack. An easier way to do this is to drill pilot holes on each of the centers and ream the holes for a snug fit on the plugs when they are inserted into their respective jacks. Don't worry too much if the fit is not exact, we'll take care of that when we connect the plugs.

7) Cut the copper conductor of the printed circuit board midway between the "REM" and "AUX" plug holes. This may be accomplished by making two cuts approximately $^{1}/_{16}$" apart with a sharp knife and peeling off the conductor between the cuts. Make sure that the cut is complete and there is no conductivity between the opposite ends of the board.

8) Clean the copper side of the board and tin the entire surface.

9) Place the printed circuit board over the plugs inserted into their jacks on the CCR-81 and with the clad side of the board out.

10) *Carefully* solder the printed circuit board to the plugs. Use a soldering gun or high wattage soldering iron and do not apply heat any longer than necessary to preclude melting of the cassette tape recorder case.

11) Cut three lengths of miniature shielded cable (278-752) 17" long.

12) Prepare one end of each of these cables by stripping the cover ⅜" from the end, pull out and twist the shield, and strip the center conductor ⅛" inch from the end.

13) Tin the shield and the center conductor of each cable.

14) Connect and solder the center conductor of each cable to the center connection of the plugs. This is easily accomplished by placing the tinned cable center conductor under the short lug, bending the lug over the wire, and then soldering the connection. During this step be very careful not to "bridge" the plug center connection to the outside connection.

15) Dress the "EAR" cable along the printed circuit board away from the "REM" plug and solder the cable shield to the printed circuit board adjacent to the plug.

16) Repeat step 15 for the "AUX" and "REM" cables.

17) When you are finished with step 16, you should have all three cables fanned out from the "EAR" end of your board, the shields of the "EAR" and "AUX" cables should be soldered to the same area on the printed circuit board, and the shield of the "REM" soldered to the isolated end of the printed circuit board. At this time, mark each cable to facilitate connector hook-up, steps 24 through 28.

18) Twist the cables together. This is most easily accomplished by holding the "EAR" and "AUX" cables in each hand. Hold the cables high enough for the connectors and "REM" cable to clear the floor and then gently twist the two cables in opposite directions. After the "EAR" and "AUX" cables are together, dress the "REM" cable around them.

19) Cut all of the cables to equal length.

20) Strip the cover from each cable ¾" from the end, pull out and twist the shield, and strip the center conductor ⅛" from the end.

21) Tin the center conductors. Do not tin the shields at this time.

22) Remove the cover from a 5-pin DIN Audio/Video Plug (274-003).

23) Insert the three cables through the plug cover.

24) Referring to Figure 2, solder the center conductor of the "EAR" cable to pin 4 of the DIN plug, and the center conductor of the "AUX" cable to pin 5.

FIGURE 2.

5-PIN DIN PLUG REAR VIEW

25) Twist the shields of the "EAR" and "AUX" cables together, tin them, cut them to appropriate length, and solder them to pin 2.

26) Solder the center conductor of the "REM" cable and tin it.

27) Twist the shield of the "REM" cable and tin it.

28) Cut the "REM" cable shield to appropriate length and solder it to pin 1.

29) Inspect your connections very carefully to assure that none of the connections are "shorted."

30) Reassemble the DIN plug.

When connecting the DIN plug as outlined in steps 24 through 28, it is preferable to insulate the terminals using a ¼" length of shrink tubing on pins 3, 4, and 5 and ⅜" length on pin 1; however, if the "REM" shield is tinned adequately to provide enough "stiffness" to hold the shield away from the other connections, it is not necessary.

The parts specified above are referenced to the Radio Shack catalog and its substitute parts may be used. As an example, you will use approximately 4½ inches of shielded cable from a 30 foot roll, leaving quite a bit left over!

After you have finished this little project, you will have a new PoCo to cassette tape recorder cable which will allow you to leave the recorder attached while the system is in its briefcase.

I hope you enjoy your PoCo and that this addition will give you additional enjoyment due to its added convenience.

PCM

CREATE 'LETTER PERFECT' TEXT FILES WITH THIS

PRINT FORMAT PROGRAM

By Gary A. Sissala

Shortly after purchasing my 24K M100 I realized that printing out text files was too awkward and cumbersome. I had to connect the M100 up to a desktop computer, load the desktop's communications software, transfer the file from one computer to another, load the desktop's word processor, load the file into it and set the print parameters for printout . . . all this for only a two-page letter. The alternative was to use the print routine in the M100 which allows you to set only line length, but not page length, double or triple spacing, and left margins. To print out an acceptable letter requires all of these. For this reason I decided to write my own print format program for the M100.

First, I chose the things I wanted the program to handle: page size, number of printed lines, line length, line spacing and left margin. I wrote the program in BASIC, and when finished, it occupied just 6K of RAM storage, leaving more than enough RAM for other files.

The program is comprised of two sections. The first, lines 10—2780 and 10000—10600, deals with screen setup and input of the text's filename along with any changes to the print format parameters. The second, lines 3000—9140, handles the actual printing of the file.

When running the program, you enter the first section, where the parameters, commands, and the input line message are displayed on the LCD screen. You then have several options. You can change any of the print format parameters, enter the filename, proceed to print or end the session.

The print format parameters are page length (PL), printed lines per page (LP), line spaces (LS), line length (LL), and left margin (LM). To change a print parameter just type in the two letter code followed by the new parameter. After you press ENTER, the new parameter will appear on the display. Note that all commands and filenames must be in capital letters.

To input the filename type Control F and the input line will change to "Enter Command." To exit the program from the display type Control E.

To print the text file type Control P. The message "Printing" will appear above the input line and, after a moment's delay, the program will begin printing the file. Note that you must have a filename entered on line five of the display before printing can commence.

After Control F has been entered, the program advances to the second section of the program, which handles the actual printing of the text file. This section inputs one character at a time building the line to be printed and checks for carriage returns, tabs, and EOF which activate various routines in this section.

Special features include suppression of leading spaces, allowance for tabulated columns and advancement to the top of the next page. An error trapping routing is used to catch bad filenames. This routine will allow you to reenter the filename if an error occurs. Finally, when the program is through printing your text file, the M100 will beep once and the "Printing" message will disappear.

This program will expand the utility of the M100 for students and others who wish formated text printouts and do not have access to a desktop computer, or those who need to print out small text files and wish to avoid the inconvenience of hooking up to their desktop.

The listing:

```
10 CLS
20 '************************************
30 '    PRINT FORMAT PROGRAM FOR M100
40 '         BY GARY A. SISSALA
50 '            2732 JERSEY AVE SO
60 '            ST. LOUIS PARK,MN 55426
70 '            (612) 922-4716
80 '            JUNE 1983
90 '************************************
100 '***********************************
110 '   DEFAULT VALUES
120 '***********************************
130 PL%=66:'    PAGE LENGTH
140 LP%=50:'    LINE PRINTED PER PAGE
150 LS%=1:'     LINE SPACES
160 LL%=60:'    LINE LENGTH
170 LM%=0:'     LEFT MARGIN LENGTH
200 '***********************************
210 '   SCREEN SETUP
220 '***********************************
230 PL$="PL"+MID$(STR$(PL%),2,5)+" ":LP$
="LP"+MID$(STR$(LP%),2,5)+" ":
240 LL$="LL"+MID$(STR$(LL%),2,5)+" ":LS$
="LS"+MID$(STR$(LS%),2,5)+" ":
250 LM$="LM"+MID$(STR$(LM%),2,5)+" ":
260 PRINT@10,"PRINT FORMAT PROGRAM";
270 PRINT@40,"(PL1-66)-------":PRINT@60,
```

```
"(LP1-66)-------":
280 PRINT@80,"(LL1-80)-------":PRINT@100
,"(LS1-3)--------":
290 PRINT@120,"(LM0-20)-------":PRINT@14
0,"(CTRL-P) TO PRINT":
300 PRINT@160,"FILENAME (CTRL-F)>--":
310 PRINT@212,"(CTRL-E) TO EXIT
320 PRINT@54,PL$:PRINT@74,LP$:
330 PRINT@94,LL$:PRINT@114,LS$:
340 PRINT@134,LM$:
350 PRINT@280,"ENTER COMMAND>-------";
400 '***********************************
410 '   SET UP ARRAY
420 '***********************************
430 FOR TA%=2 TO 6
440 READ CC$(TA%)
450 NEXT TA%
460 RESTORE:GOTO 1000
500 DATA "PL","LP","LL","LS","LM"
1000 '**********************************
1010 '  PROCESSING ANCHOR
1020 '**********************************
1030 GOSUB 10000
2000 '**********************************
2010 '  COMMAND CHECK
2020 '**********************************
2030 PRINT@260,SPACE$(15);
2040 FOR AP%=2 TO 6
2050 FL%=1
2060 IF MID$(CL$,1,2)=CC$(AP%) THEN FL%=
AP%
2070 ON FL% GOTO 2080,2200,2300,2400,250
0,2600
2080 NEXT AP%
2090 PRINT@240,"  SYNTAX ERROR-PRESS ENT
ER TO CONTINUE"
2100 CL$=""
2110 E$=INKEY$:IF E$="" THEN GOTO 2110
2120 IF E$=CHR$(13) THEN GOTO 2130 ELSE
GOTO 2110
2130 PRINT@240,SPACE$(39);
2140 PRINT@301,SPACE$(15);
2150 GOTO 1030
2170 '**********************************
2180 '  COMMAND PROCESSING
2190 '**********************************
2200 T$=MID$(CL$,3,2)
2210 IF VAL(T$)<1 OR VAL(T$)>66 THEN GOT
O 2090
2220 PL%=VAL(T$)
2230 PL$="PL"+MID$(STR$(PL%),2,5)+" ":
2240 PRINT@54,PL$:
2250 CL$=""
2260 PRINT@301,SPACE$(10);
2270 GOTO 1030
2300 T$=MID$(CL$,3,2)
2310 IF VAL(T$)<1 OR VAL(T$)>66 THEN GOT
O 2090
2320 LP%=VAL(T$)
2330 LP$="LP"+MID$(STR$(LP%),2,5)+" "
2340 PRINT@74,LP$:
2350 CL$=""
2360 PRINT@301,SPACE$(10);
2370 GOTO 1030
2400 T$=MID$(CL$,3,2)
2410 IF VAL(T$)<1 OR VAL(T$)>80 THEN GOT
O 2090
2420 LL%=VAL(T$)
2430 LL$="LL"+MID$(STR$(LL%),2,5)+" "
2440 PRINT@94,LL$:
2450 CL$=""
2460 PRINT@301,SPACE$(10);
2470 GOTO 1030
2500 T$=MID$(CL$,3,1)
2510 IF VAL(T$)<1 OR VAL(T$)>3 THEN GOTO
 2090
2520 LS%=VAL(T$)
2530 LS$="LS"+MID$(STR$(LS%),2,5)+" ":
2540 PRINT@114,LS$:
2550 CL$=""
2560 PRINT@301,SPACE$(10);
2570 GOTO 1030
2600 T$=MID$(CL$,3,2)
2610 IF VAL(T$)<0 OR VAL(T$)>20 THEN GOT
O 2090
2620 LM%=VAL(T$)
2630 LM$="LM"+MID$(STR$(LM%),2,5)+" ":
2640 PRINT@134,LM$:
2650 CL$=""
2660 PRINT@301,SPACE$(10);
2670 GOTO 1030
2700 PRINT@280,"Enter Filename>------";
2710 GOSUB 10000
2720 FN$=CL$
2730 CL$=""
2740 PRINT@301,SPACE$(15);
2750 PRINT@180,SPACE$(15);
2760 PRINT@180,FN$;
2770 PRINT@280,"ENTER COMMAND>-------";
2780 GOTO 1030
3000 '**********************************
3010 'ZERO VARIABLES AND OPEN FILE
3020 '**********************************
3030 AL%=0:TF%=0:IS%=0:T%=0:B$="":IS$=""
:NL%=0:
3040 ON ERROR GOTO 9000
3050 OPEN FN$ FOR INPUT AS 1
3060 PRINT@240,SPACE$(39);
3070 PRINT@240,"PRINTING";
3100 '**********************************
3110 'INPUT DATA CHARACTER AND CHECK
3120 '  FOR EOF,TAB,ENTER,LEADING
3130 '  SPACES AND LINE LENGTH
3140 '**********************************
3150 ON ERROR GOTO 12000
3200 IF NOT EOF(1) THEN IS$=INPUT$(1,1)
ELSE GOTO 6300
3210 IF IS$=CHR$(13) THEN GOTO 3500
3220 IF IS$=CHR$(9) THEN TF%=TF%+(8-((AL
%+TF%)MOD8))
```

```
3230 IF AL%=0 AND IS$=CHR$(32) THEN GOTO
 3200
3240 B$=B$+IS$:AL%=AL%+1
3250 IF (AL%+TF%)=>(LL%+1) THEN GOTO 400
0 ELSE GOTO 3200
3500 IS$=INPUT$(1,1)
3510 IF AL%=0 THEN LPRINT ELSE GOTO 4000
3520 GOTO 6000
4000 '*********************************
4010 '  LINE LPRINT
4020 '*********************************
4030 IF LEN(B$)=0 THEN GOTO 6300
4040 IF AL%+TF%<=LL% THEN GOSUB 4500 ELS
E GOTO 4100
4050 GOTO 6000
4100 '***** FIND LAST SPACE IN B$ *****
4110 SP%=0
4120 FOR T%=LEN(B$) TO 1 STEP -1
4130 SP%=INSTR(T%,B$,CHR$(32))
4140 IF SP%=0 THEN NEXT T% ELSE T%=0
4150 IF SP%=0 THEN GOSUB 4500 ELSE GOSUB
 4700
4160 IF LEN(B$)=0 THEN GOTO 6000 ELSE GO
TO 5000
4500 '***** NO SPACE ******************
4510 IF LEN(B$)>LL% THEN GOTO 4560
4520 LPRINT SPACE$(LM%)+B$
4530 B$="":AL%=0:TF%=0:
4540 RETURN
4560 LPRINT SPACE$(LM%)+MID$(B$,1,(LL%-T
F%))
4570 B$=MID$(B$,((LL%-TF%)+1),(LEN(B$)-(
LL%-TF%)))
4580 AL%=LEN(B$)
4590 RETURN
4700 '***** FOUND SPACE ***************
4710 LPRINT SPACE$(LM%)+MID$(B$,1,(SP%-1
))
4720 IF SP%<(LL%+1) THEN B$=MID$(B$,(SP%
+1),(LEN(B$)-SP%)) ELSE B$=""
4730 AL%=LEN(B$)
4740 RETURN
5000 '*********************************
5010 '  CHECK FOR TABS IN NEW B$
5020 '*********************************
5100 T%=0:TF%=0:PT%=0:
5110 T%=INSTR((PT%+1),B$,CHR$(9))
5120 IF T%=0 THEN GOTO 6000
5130 TF%=TF%+(8-((AL%+TF%)MOD8)):PT%=T%:
5140 GOTO 5110
6000 '*********************************
6010 'EOF CHECK WHICH EITHER BRANCHES
6020 '  BACK TO INPUT DATA CHARACTER
6030 '  OR PRINTS CONTENTS OF B$.  IT
6040 '  ALSO PERFORMS LINE SPACING AND
6050 '  PAGE BREAKS.
6060 '*********************************
6100 '***** LINE SPACING **************
6110 NL%=NL%+1
6120 IF LS%=1 THEN GOTO 6200
6130 FOR T%=1 TO (LS%-1)
6140 LPRINT
6150 NL%=NL%+1
6160 NEXT T%
6200 '***** PAGE BREAK ****************
6210 IF NL%<LP% THEN GOTO 6300
6220 FOR T%=1 TO (PL%-NL%)
6230 LPRINT
6240 NEXT T%
6250 NL%=0
6300 '***** CHECK FOR EOF *************
6310 IF NOT EOF(1) THEN GOTO 3200
6320 IF LEN(B$)>0 THEN LPRINT SPACE$(LM%
)+B$ ELSE GOTO 6340
6330 NL%=NL%+1:B$="";
6340 IF NL%=0 THEN GOTO 6380
6350 FOR T%=1 TO (PL%-NL%)
6360 LPRINT
6370 NEXT T%
6380 PRINT@240,SPACE$(39)
6390 BEEP:CLOSE:GOTO 1030
9000 '*********************************
9010 ' FILENAME ERROR
9020 '*********************************
9030 IF ERR=52 OR ERR=53 OR ERR=55 THEN
9100
9040 PRINT "ERROR ";ERR;"  AT LINE NUMBE
R ";ERL:
9050 END
9100 PRINT@240,"Invalid filename press E
NTER to contine";
9110 E$=INKEY$:IF E$="" THEN GOTO 9110
9120 IF E$=CHR$(13) THEN GOTO 9130 ELSE
GOTO 9110
9130 PRINT@240,SPACE$(39);
9140 RESUME 2700
10000 '*********************************
10010 '       INPUT COMMAND
10015 '*********************************
10020 CC%=0
10030 IC$=INKEY$:IF IC$="" THEN 10030
10040 IF IC$=CHR$(29) OR IC$=CHR$(8) THE
N GOTO 10500
10045 IF IC$=CHR$(5) THEN GOTO 10600
10050 IF IC$=CHR$(13) THEN RETURN
10060 IF IC$=CHR$(6) THEN GOTO 2700
10070 IF IC$=CHR$(16) THEN GOTO 3000
10080 CC%=CC%+1
10090 PRINT@(300+CC%),IC$;
10100 CL$=CL$+IC$
10110 IF CC%=13 THEN RETURN
10120 GOTO 10030
10500 '***** BACKSPACE ****************
10510 IF CC%<=0 THEN GOTO 10030
10520 PRINT@(300+CC%)," ";
10530 CC%=CC%-1:CL$=MID$(CL$,1,CC%)
10540 GOTO 10030
10600 CLS:END
12000 PRINT ERL
12010 PRINT ERR:END
```

Reviews

SOFTWARE

Scribe Useful Word Processor With Unexpected Power

Scribe is the first word processing package developed for the Model 100. Its purpose is to allow the user to produce a file and then utilize the full capabilities of the available printer. In my job as Newsletter Editor for the Cincinnati TRS-80 Users Group (CINTUG), I have experimented with all of the word processors written for the Models I, II/12/16, and III/4. Each of the packages has features that fit a users specific need and printer, Chattanooga Systems Associates has chosen to model their program so that it is compatible with the *Newscript* word processing package. This particular system follows the Edgar word processor developed by IBM. With this compatibility, *Scribe* possesses considerable more power than might be expected from a Model 100 word processor. Text can be prepared on a Model 100, and then transmitted to a Model I/III/4, to be merged with the mailing list for automatic addressing.

Scribe, like *Newscript*, supports a variety of printers. It allows you to add imbedded control codes so that double wide, 10 cpi, and 16.7 cpi characters can be included in a single printout. The program itself is a short BASIC program some 47 lines in length, which requires only 2.2K of memory. This leaves a little more than 3K for the text file in an 8K machine. The original version that you receive is set up for use with three printers, the Epson, Gemini 10/15 and RS DW-II (which is the current standard for all Radio Shack Printers). Information is included so that you can modify the program for a different printer.

In order to use *Scribe*, you first prepare your text using the *TEXT* program in the Model 100. This text preparation should include the formatting described in the *Scribe* manual. You then return to BASIC and load up either from tape or from RAM. *Scribe* then asks a series of questions. First, it asks where the text file is to be found, RAM or CAS. It then asks the name of the file. The third question has to do with format . . . 10 cpi, 12 cpi, or 17 cpi. Then it asks for the print quality that you want (providing your printer supports these modes). Next it asks for the desired line spacing. Automatic page numbering is the next question to be answered along with the page where numbering is to start. Finally, you are asked if it is to stop at the end of each page so that a new sheet of paper can be inserted. This feature is useful if your printer allows you to friction-feed single sheets.

The best way to understand the use of *Scribe* is to type in the sample letter given in the manual. After doing this, I had no problems using *Scribe* with both the EPSON and GEMINI printers. It also worked well with my Centronics 737/LPIV, with the exception that the line feed command did not work with-

The Rackseller

ALABAMA	
Florence	Anderson News Co.
Madison	Madison Books
ALASKA	
Fairbanks	Electronic World
	Fairbanks News Agency
Kotzebue	Arctic Bookstore, Inc.
ARIZONA	
Phoenix	Road Runner
	Computer Pro
Scottsdale	Softwareland Corp.
Tucson	Anderson News Co.
CALIFORNIA	
Atwater	Electro Brain
Citrus Heights	Software Plus
Half Moon Bay	Strawflower Electronics
North Hollywood	Levity Distributors
Sunnyvale	Computer Literacy
CONNECTICUT	
Hartford	The Aetna Life Club Store
DELAWARE	
Wilmington	Normar Inc.—The Smoke Shop
FLORIDA	
Ft. Lauderdale	Software Connection
Miami	Newsrack
Panama City	Computer Systems Group
Pensacola	Anderson News Co.
Tallahassee	Anderson News Co.
Winter Park	Independent Computer Systems
GEORGIA	
Atlanta	Guild News Agency
Columbus	Muscogee News Co.
Trenton	The Phone Line
ILLINOIS	
Chicago	Kroch's & Brentano's Jackson Street
	Kroch's & Brentano's Wabash Ave.
Chicago	Prairie News Agency
Oakbrook	Kroch's & Brentano's
INDIANA	
Elkhart	Elkhart City News & Book Store
Indianapolis	A—Computer Store
Madison	Arco Office Supplies
Scottsburg	Radio Shack of Scottsburg
KANSAS	
Topeka	Town Criers of Topeka
	Palmer News Inc.
Wichita	Amateur Radio Equipment Co.
	Lloyd's Radio
KENTUCKY	
Benton	Anderson Electronics
Pikeville	Gus-Stan Enterprises
LOUISIANA	
Slidell	Radio Shack
MAINE	
South Portland	Portland News Co.
MASSACHUSETTS	
Littleton	Computer Plus
Marlboro	Radio Shack
Woburn	Microcon Software Centers
MICHIGAN	
Novi	MI Software Dist., Inc.
NEBRASKA	
Lincoln	Hobby Town
NEVADA	
Las Vegas	Hurley Electronics
NEW HAMPSHIRE	
Manchester	Brookwrights
Petersborough	Radio Shack
West Lebanon	Verham News Corp.
NEW JERSEY	
Hackensack	Total Circulation Service
Midland Park	Software City
Pennsville	Dave's Electronic Radio Shack
NEW MEXICO	
Albuquerque	Page One Newsstand
NEW YORK	
Johnson City	Unicorn Electronics
New York City	Usercom System, Inc.
NORTH DAKOTA	
Fargo	Computer Associates
OHIO	
Cincinnati	Softwarehouse
Lakewood	Lakewood International News
Toledo	Leo's
	Reitz Electronics, Inc.
OREGON	
Eugene	Libra Books, Inc.
Medford	John's News Stand
Portland	80-Plus
Roseburg	Rainy Day Bookshop
PENNSYLVANIA	
Pleasant Hills	Pittsburgh Computer Store
SOUTH CAROLINA	
Greenville	Palmetto News Co.
North Charleston	The Green Dragon
TENNESSEE	
Chattanooga	Anderson News Co.
Knoxville	Anderson News Co.
Memphis	Computer Center
Nashville	Mosko's Book Store
Smyrna	Delker Electronics, Inc.
TEXAS	
Austin	Austin News Agency
	Capitol Micro Computers
	Software & Things
Dallas	Micro Concepts, Inc.
Elgin	Homing Pigeon
Ft. Worth	A & A International
	RFI Electronics
Irving	Software Access
WASHINGTON	
Blaine	Compulit Dist.
Richland	C & J Electronics Computer Center
Seattle	Adams News
WISCONSIN	
Appleton	Badger Periodicals
Janesville	Book World
Portage	Sam's Electronics
CANADA:	
ALBERTA	
Calgary	Rainbow Software Services
Edmonton	Kelly's Software Distributors
NOVA SCOTIA	
Halifax	Atlantic News
ONTARIO	
Hamilton	Galls Book World

out modifying the program. I was able to modify the program to work with the Centronics 737 by using the information supplied in the manual. I also uploaded the file to my Model I, and printed it out using *Newscript* with no problems.

My major complaint with *Scribe* is the manual supplied. It was printed on folded 7½ by 11 paper, and the copy had been reduced approximately 40% in reproduction. This size is too small to properly file in a note book. The reduced type size also made it difficult to read, and I made a number of errors in typing the sample letter. Other than the quality of the documentation reproduction, I found no major faults with the program. It should prove useful for those occasions when a text file cannot be uploaded to a larger machine.

(Chattanooga Systems Associates, P.O. Box 22261, Chattanooga, TN 27421, $24.95)

—Ed Fairman

(continued from page 5)

the Portable Computer are making good progress. So, effective this month, there is a special edition of *PCM* available in Australia, New Zealand and the South Seas.

And, in closing, let me call your attention to a computer show — one of four — that we will be sponsoring October 14-16 in Fort Worth, Texas. The show, *RAINBOWfest*, is essentially aimed at the TRS-80 Color Computer, but last year's first show in Chicago saw a couple of Portable Computers around. I suspect there will be more this year.

Too, our keynote speaker at a Saturday morning breakfast is Mark Yamagata, who has just been appointed Director of Computer Merchandising for Personal Computer Products. That area of responsibility includes the Portable Computer as well as the Color Computer. It will be one of Mark's first public appearances in his new role.

Too, *RAINBOWfest* might be a glimpse into the PoCo's future. After all, the Color Computer is a little more than two years older than the Portable Computer. And, I think you will be amazed to see everything that is there!

The site is the Hyatt-Regency Fort Worth. The display floor is open Friday evening, most of the day Saturday and all afternoon on Sunday. Three day tickets are $8 in advance while single day tickets are $6 in advance. Breakfast tickets, which must be ordered in advance, are $11. There is a $1 handling charge *per order*.

Tickets can be ordered from *RAINBOWfest*, P.O. Box 209, Prospect, KY 40059. Or, you can call (502) 228-4492. We accept VISA, MasterCard and American Express, of course.

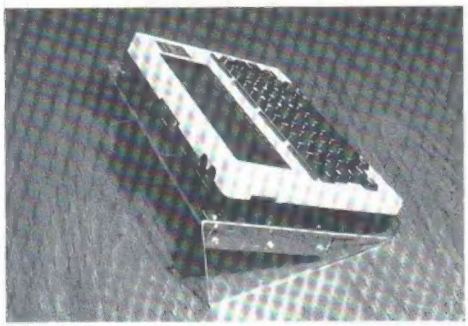
Other shows later in the year will be in Long Beach, California, Chicago and New Brunswick, New Jersey.

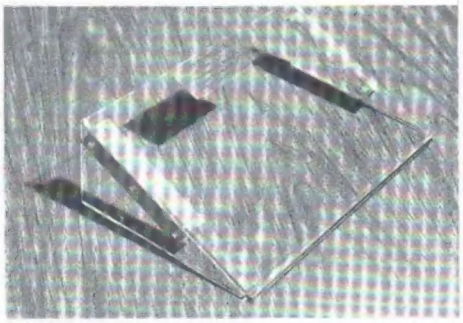
We'd like to see you there!

—Lonnie Falk

NEW!

The Spectrum Projects PoCo Stand





$24.95

plus $5 shipping and handling
N.Y. residents add sales tax

Model 100 Cables

CoCo to Model 100 Cables — $19.95

CoCo to Model I, II, III, IV — $24.95

External 6 volt Battery — $19.95

8K Model 100
stock #26-3801
$679

For quick delivery call:

Model 100 8K Memory Upgrade Module
$85 each
2 or more $80 each

SPECTRUM PROJECTS

93-15 86th DRIVE
WOODHAVEN, NY 11421
(212) 441-2807

PCM CERTIFICATION SEAL

Call our Bulletin Board for free Model 100 programs.
Now on line . . . 20 MEG. CORVUS HARD DISK DRIVE

BBS #1 — (212) 441-3755
BBS #2 — (212) 441-3766
BBS #3 — (212) 441-5719

THE SECRETS OF PERFECT MEMORY: ONE AND ONE HALF EARTH DOLLARS

AT LAST: THE WHOLE TRUTH ABOUT FLOPPIES.

Amazing book reveals all!

How to keep from brainwashing your disk so it never loses it's memory.

How fingerprints can actually damage disks. Unretouched Kirlian photographs of UFO's (Unidentified Floppy Objects)! The incredible importance of making copies: the Department of Redundancy Department—and what goes on when it goes on! Powerful secret methods that scientists claim can actually prevent computer amnesia! All this, and much more . . .

In short, it's an 80-page plain-English, graphically stunning, pocket-sized definitive guide to the care and feeding of flexible disks.

For The Book, ask your nearest computer store that sells Elephant™ disks, and bring along one and one half earth dollars.

For the name of the store, ask us.

ELEPHANT MEMORY SYSTEMS® Marketed exclusively by Leading Edge Products, Inc., Information Systems and Supplies Division, 55 Providence Highway, Norwood, MA 02062. Call toll free 1-800-343-8413, In Massachusetts, call collect (617) 769-8150, Telex 951-624.